New Valid Inequalities for the one-commodity Pickup-and-Delivery Travelling Salesman Problem

Hipólito Hernández-Pérez   Juan-José Salazar-González

DEIOC, University of La Laguna, Tenerife, Spain
hhperez@ull.es, jjsalaza@ull.es

- Let us consider a *depot*, denoted by 1, and a set of customers $\{2, \ldots, n\}$.

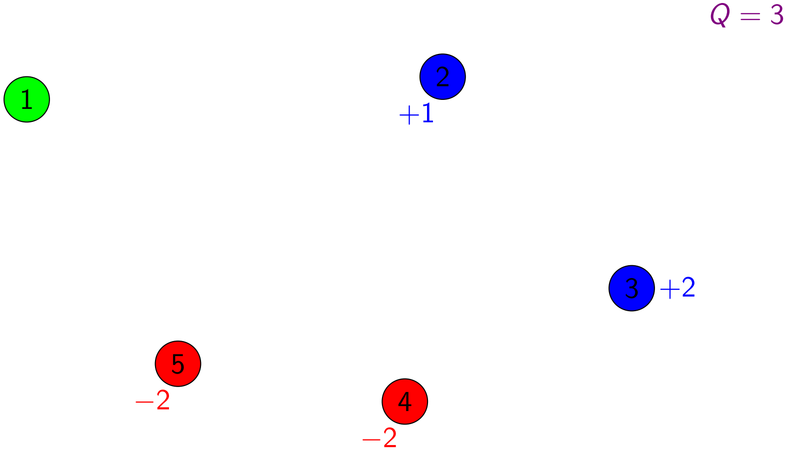
- Let us consider a *depot*, denoted by 1, and a set of customers $\{2, \ldots, n\}$.
- Each customer $i \in \{2, \ldots, n\}$ has associated a given quantity q_i of product:
  - If $q_i < 0$, then i is called *delivery customer*.
  - If $q_i > 0$, then i is called *pickup customer*.

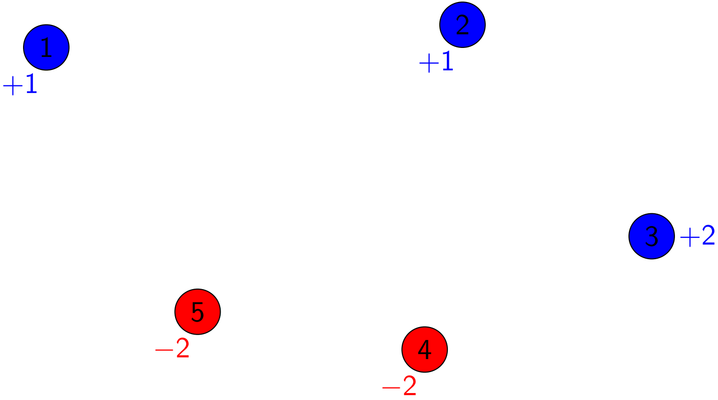
- Let us consider a *depot*, denoted by 1, and a set of customers $\{2, \ldots, n\}$.
- Each customer $i \in \{2, \ldots, n\}$ has associated a given quantity q_i of product:
  - If $q_i < 0$, then i is called *delivery customer*.
  - If $q_i > 0$, then i is called *pickup customer*.
- There is a vehicle with upper-limited *capacity* Q, originally stationed in the depot.

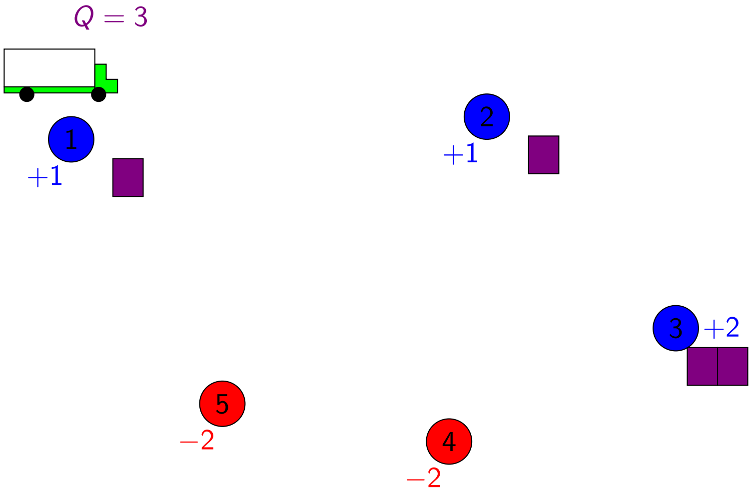
- Let us consider a *depot*, denoted by 1, and a set of customers $\{2, \ldots, n\}$.
- Each customer $i \in \{2, \ldots, n\}$ has associated a given quantity q_i of product:
  - If $q_i < 0$, then i is called *delivery customer*.
  - If $q_i > 0$, then i is called *pickup customer*.
- There is a vehicle with upper-limited *capacity* Q, originally stationed in the depot.
- There is only ONE product (e.g., money)
  NOT two commodities (full and empty bottles!).

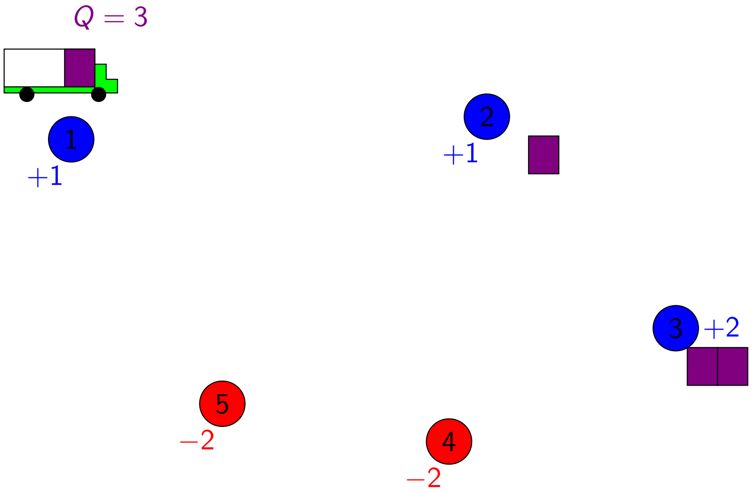
- Let us consider a *depot*, denoted by 1, and a set of customers $\{2, \ldots, n\}$.
- Each customer $i \in \{2, \ldots, n\}$ has associated a given quantity q_i of product:
  - If $q_i < 0$, then i is called *delivery customer*.
  - If $q_i > 0$, then i is called *pickup customer*.
- There is a vehicle with upper-limited *capacity* Q, originally stationed in the depot.
- There is only ONE product (e.g., money)
  NOT two commodities (full and empty bottles!).
- The travel cost c_{ij} is also given.

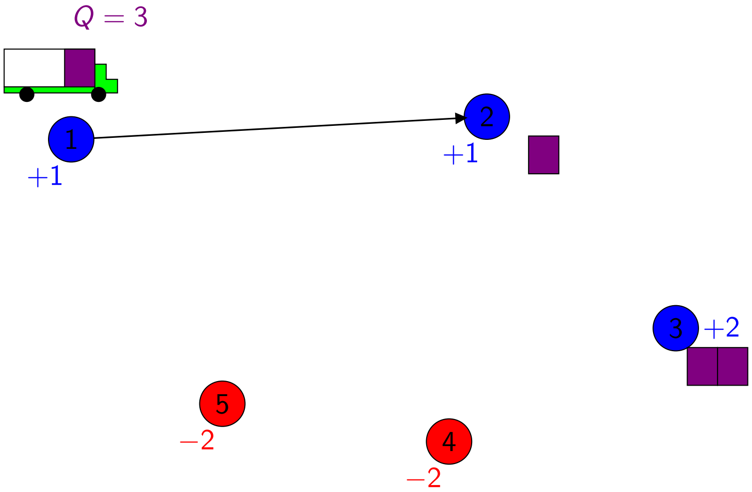
- Let us consider a *depot*, denoted by 1, and a set of customers $\{2, \ldots, n\}$.
- Each customer $i \in \{2, \ldots, n\}$ has associated a given quantity q_i of product:
  - If $q_i < 0$, then i is called *delivery customer*.
  - If $q_i > 0$, then i is called *pickup customer*.
- There is a vehicle with upper-limited *capacity* Q, originally stationed in the depot.
- There is only ONE product (e.g., money)
  NOT two commodities (full and empty bottles!).
- The travel cost c_{ij} is also given.
- No condition on the initial load of the vehicle at the depot.


$Q = 3$
1
2
+1
3
+2
5
−2
4
−2


1
+1
2
+1
3
+2
5
−2
4
−2

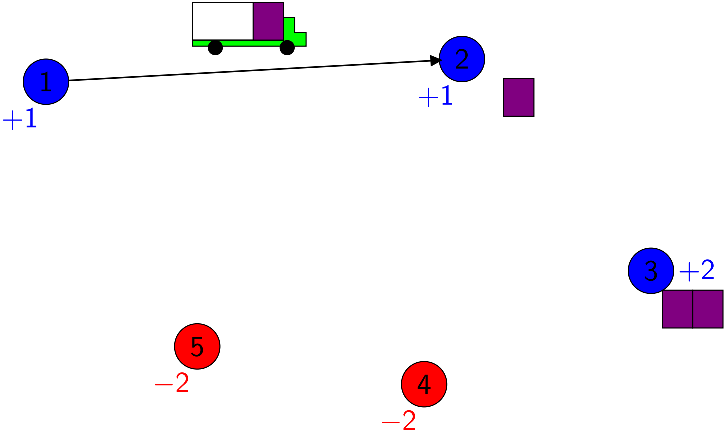

$Q = 3$
1
+1
2
+1
3
+2
5
−2
4
−2

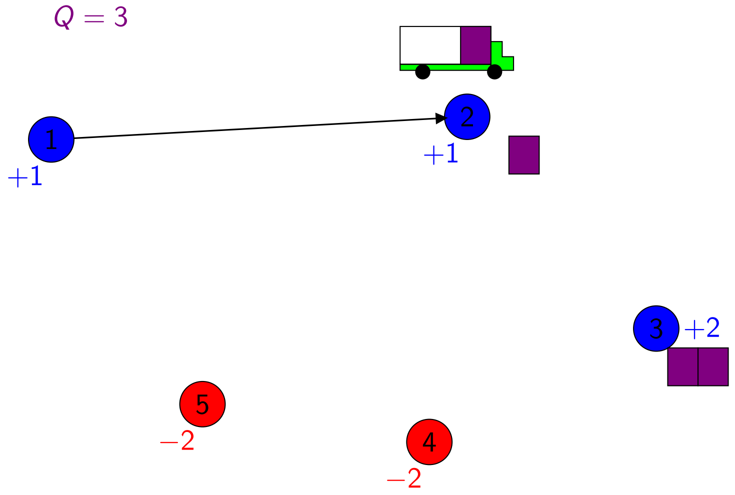

$Q = 3$
1
+1
2
+1
3
+2
4
−2
5
−2

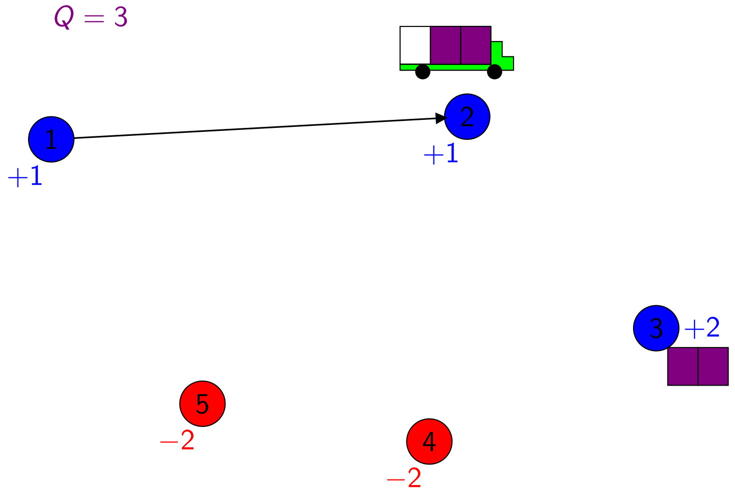


$Q = 3$
1
+1
2
+1
3
+2
4
−2
5
−2

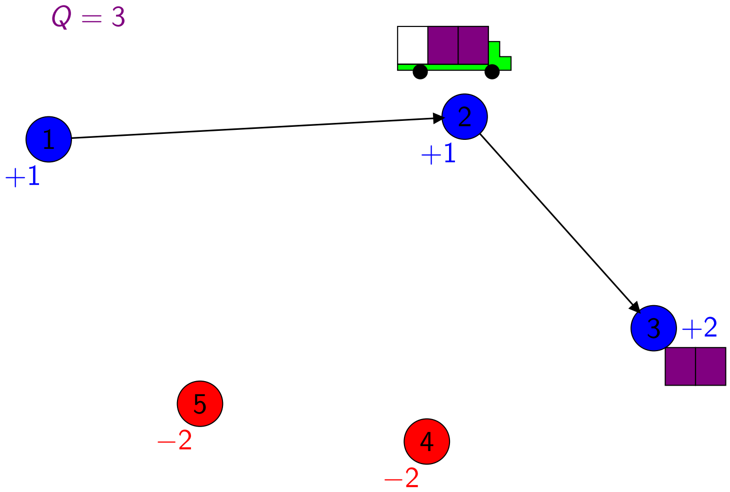
$Q = 3$

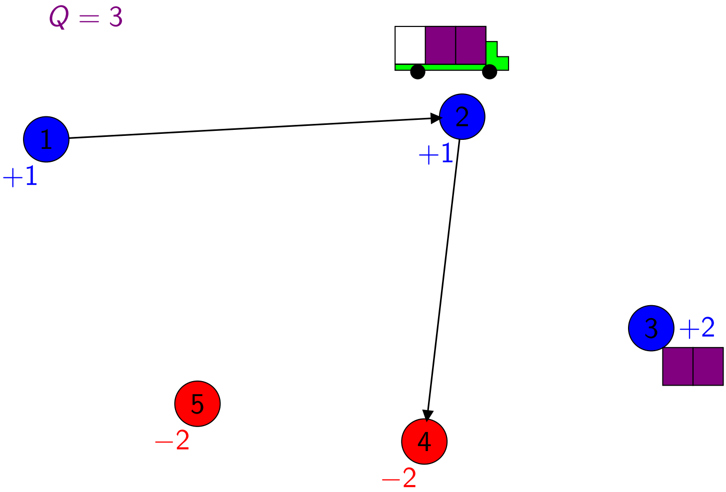



$Q = 3$
1
+1
2
+1
3
+2
4
−2
5
−2

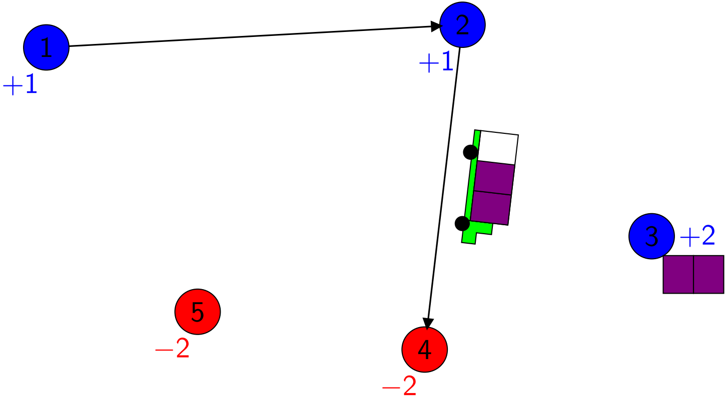


$Q = 3$
1
+1
2
+1
3
+2
4
−2
5
−2

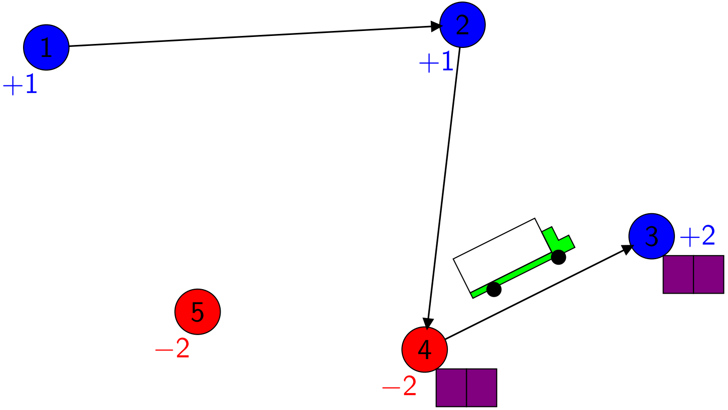

$Q = 3$
1
+1
2
+1
3
+2
4
−2
5
−2


$Q = 3$
1
+1
2
+1
3
+2
4
−2
5
−2

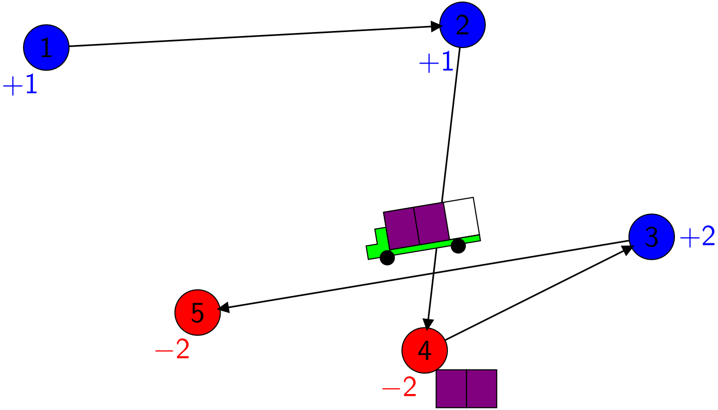
$Q = 3$



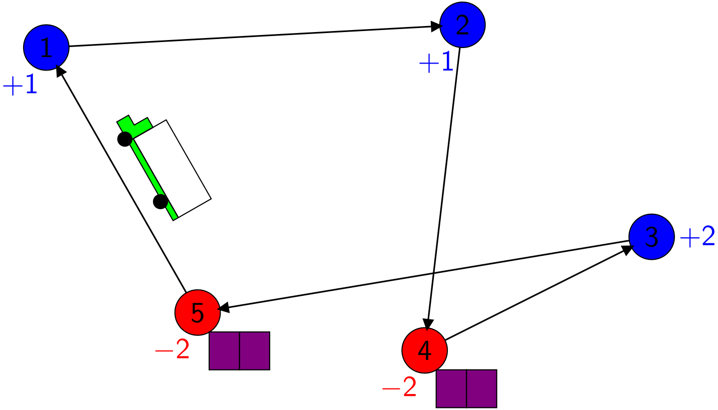
$Q = 3$

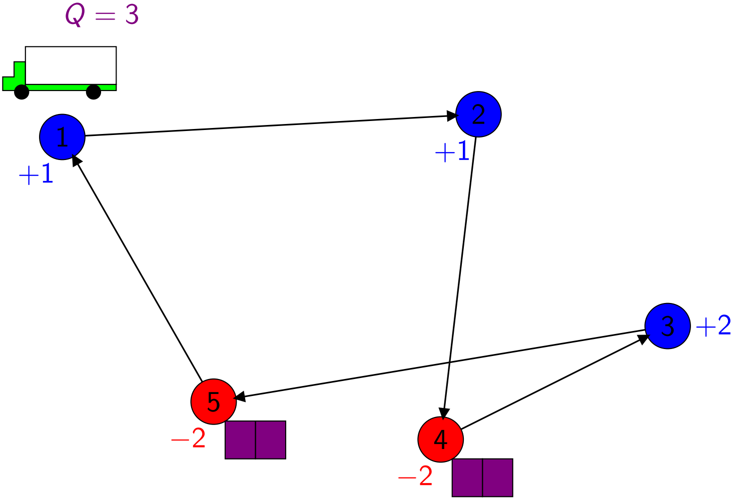


$Q = 3$

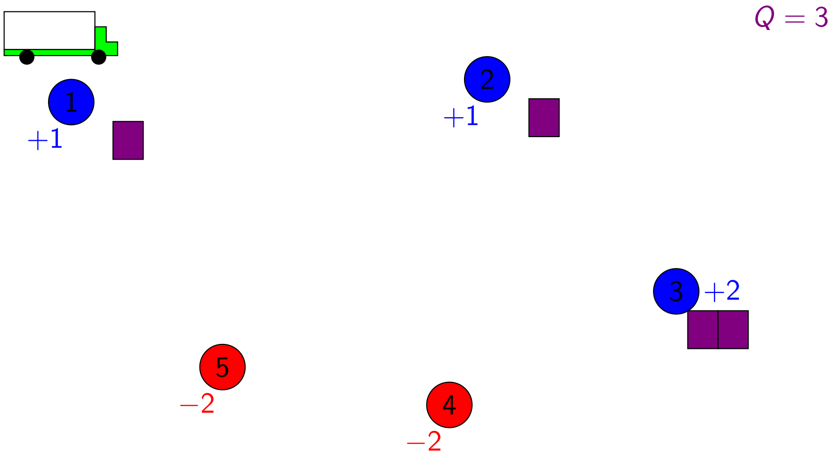


$Q = 3$

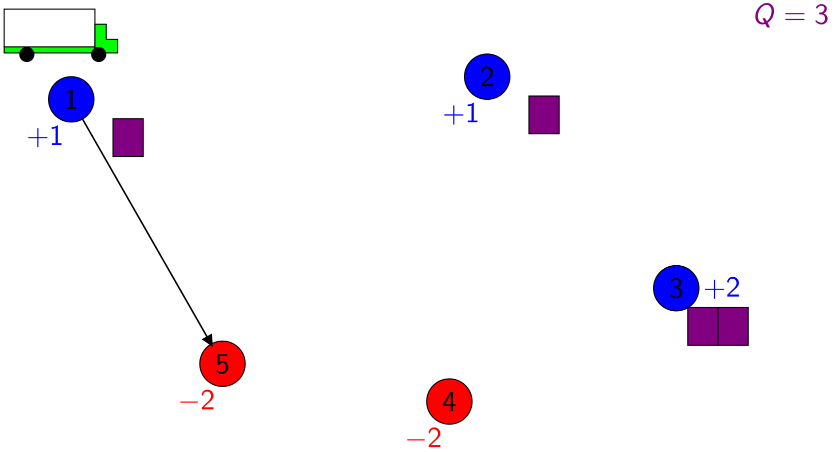



$Q = 3$
1
+1
2
+1
3
+2
4
−2
5
−2

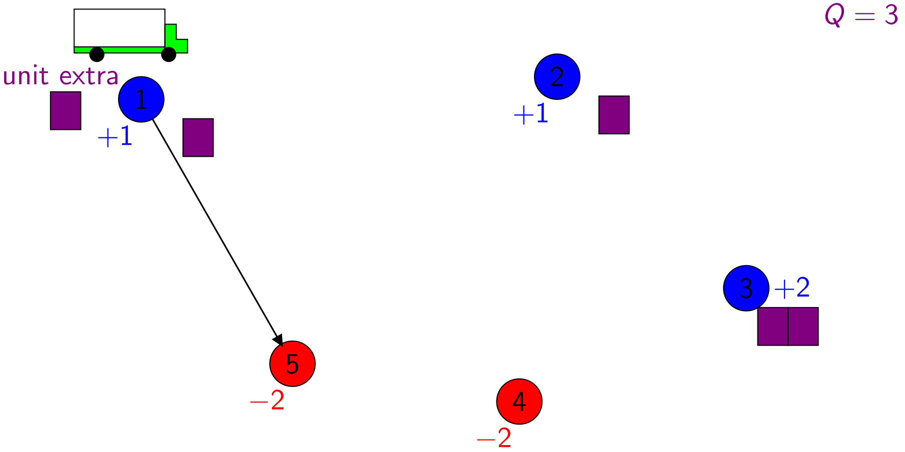
Remember: the depot can provide extra-load to the vehicle



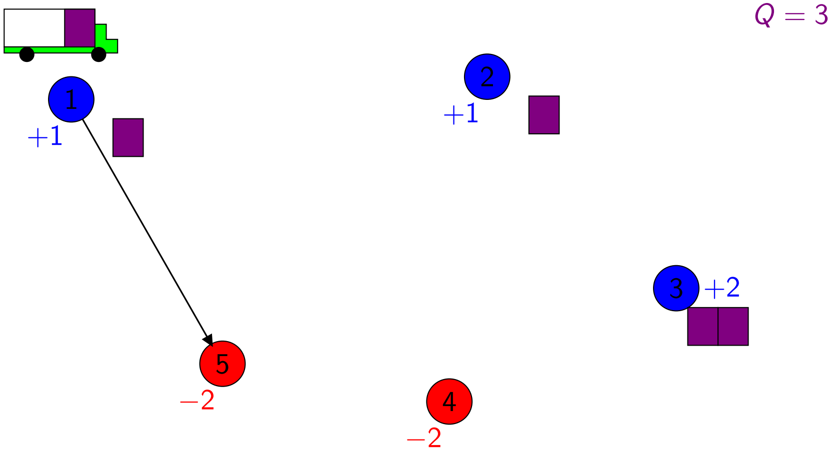
Remember: the depot can provide extra-load to the vehicle



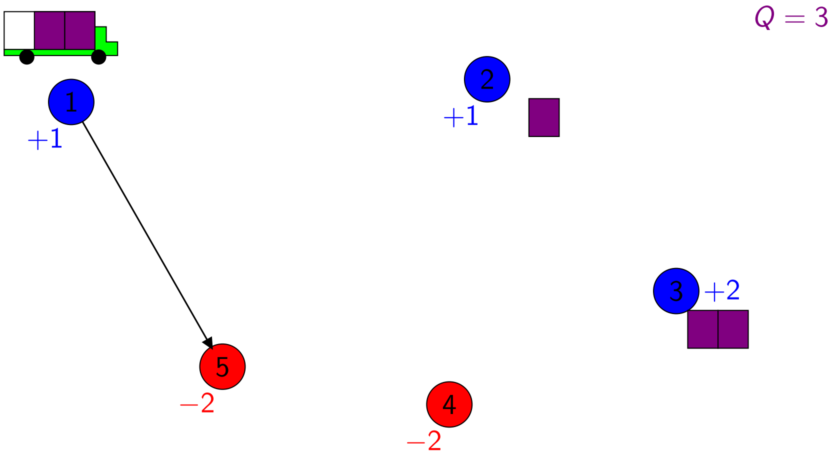
Remember: the depot can provide extra-load to the vehicle



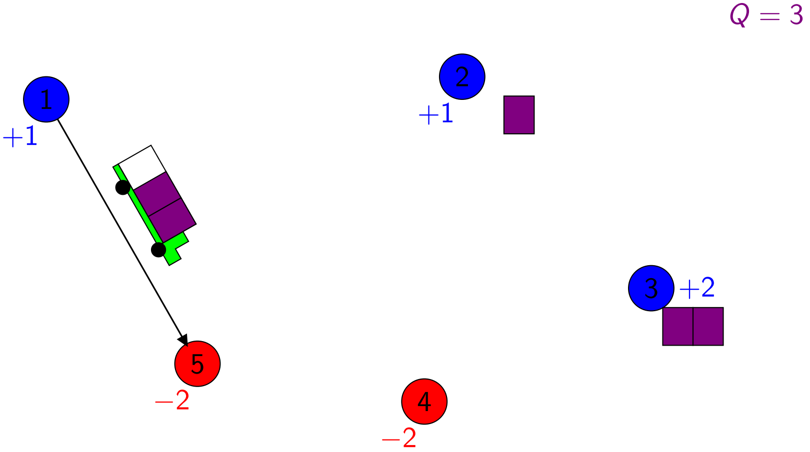
Remember: the depot can provide extra-load to the vehicle



Remember: the depot can provide extra-load to the vehicle

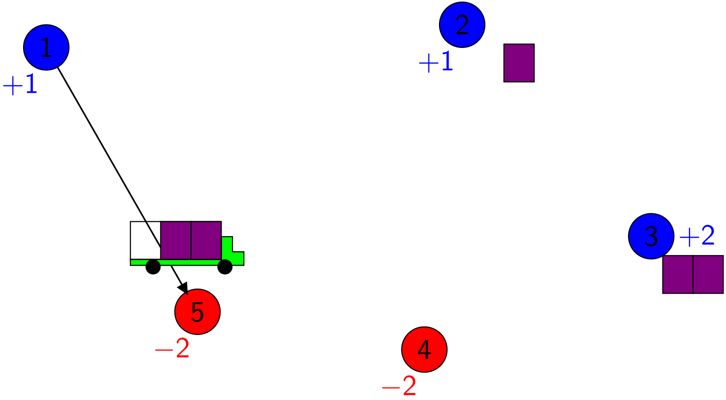


Remember: the depot can provide extra-load to the vehicle

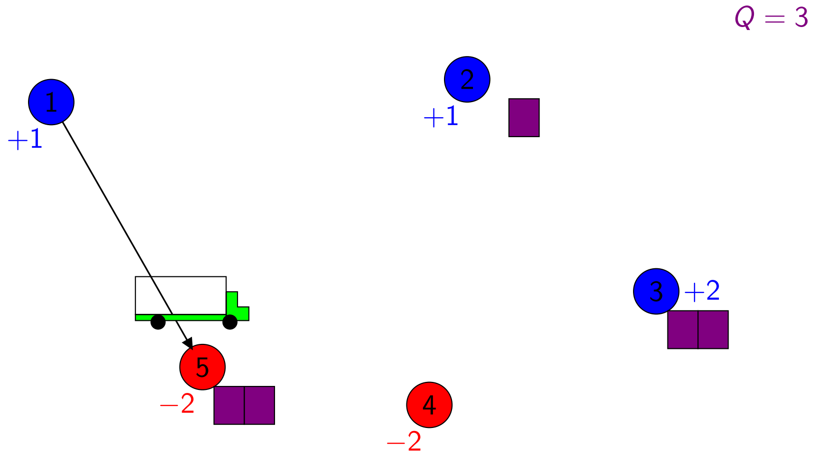


Remember: the depot can provide extra-load to the vehicle

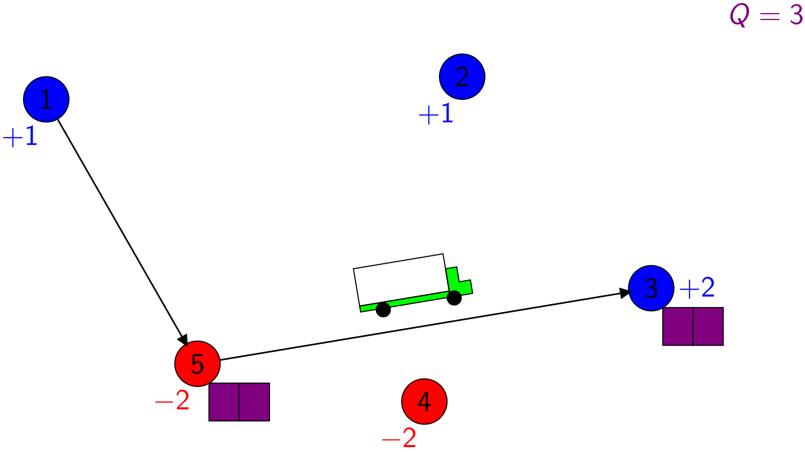
$Q = 3$



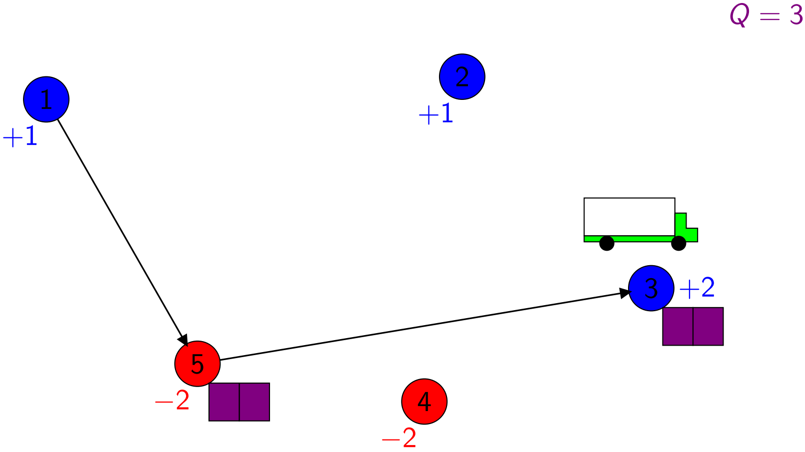
Remember: the depot can provide extra-load to the vehicle



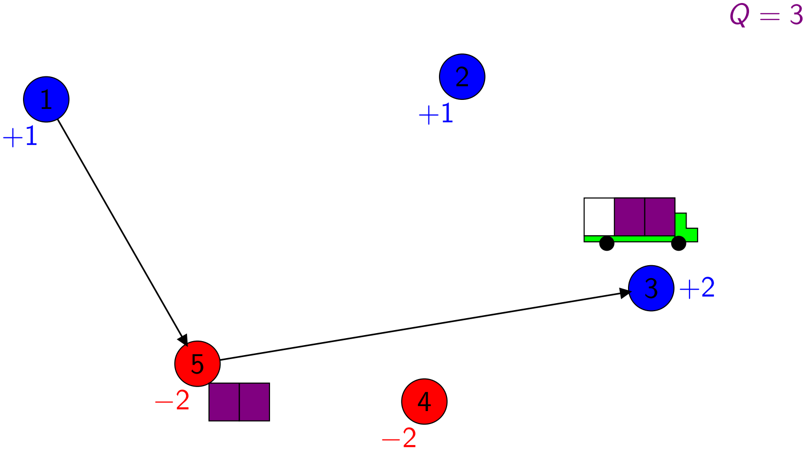
Remember: the depot can provide extra-load to the vehicle



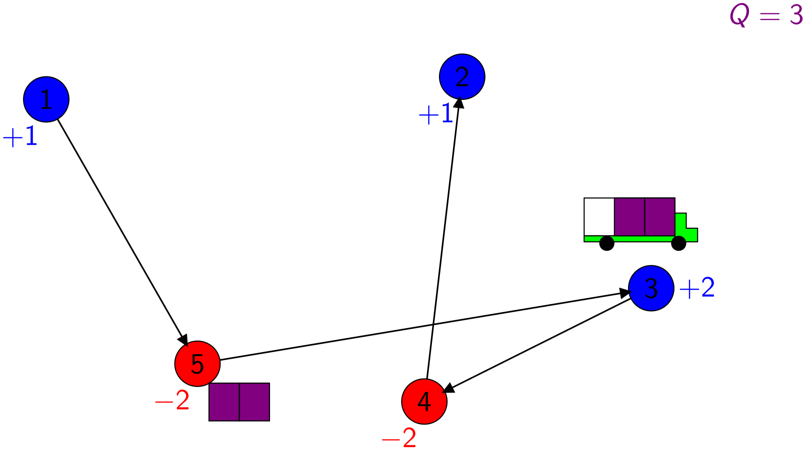
Remember: the depot can provide extra-load to the vehicle



Remember: the depot can provide extra-load to the vehicle

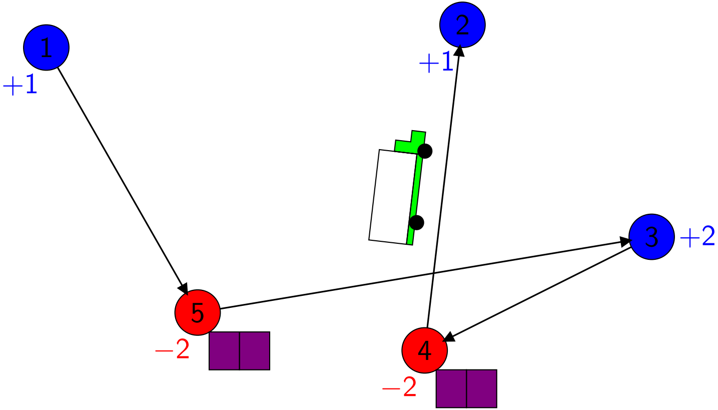


Remember: the depot can provide extra-load to the vehicle

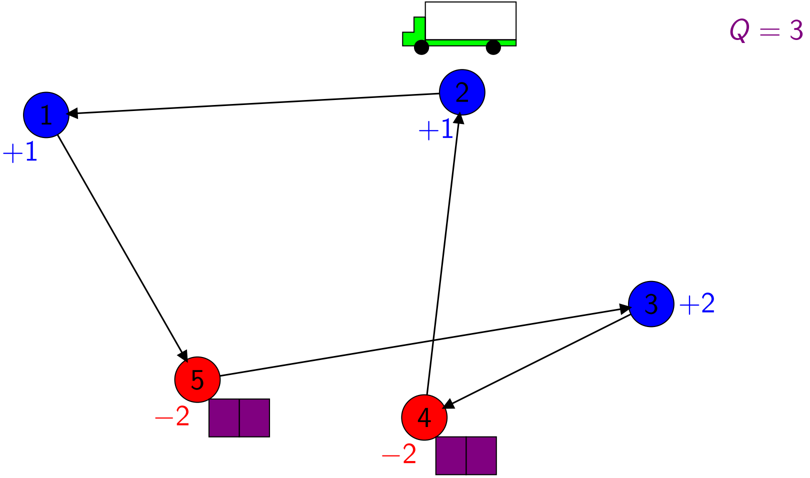


Remember: the depot can provide extra-load to the vehicle

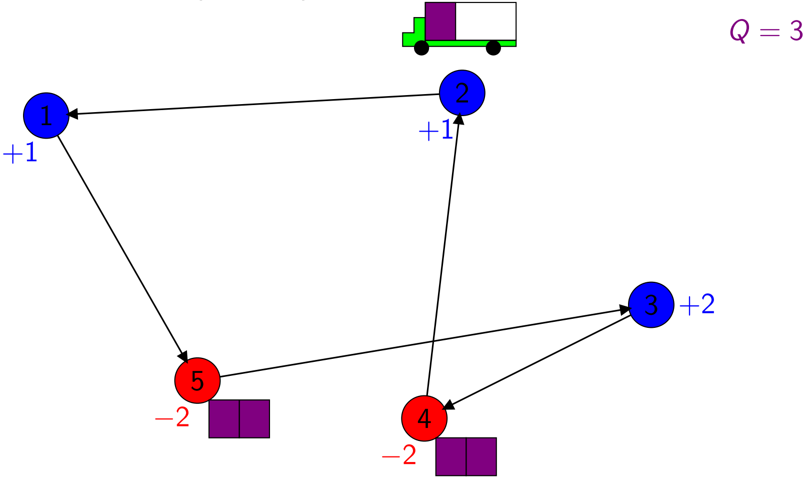
$Q = 3$



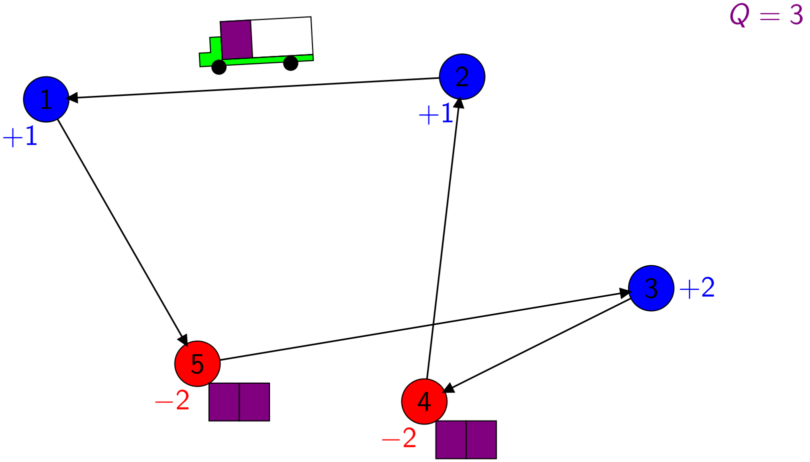
Remember: the depot can provide extra-load to the vehicle



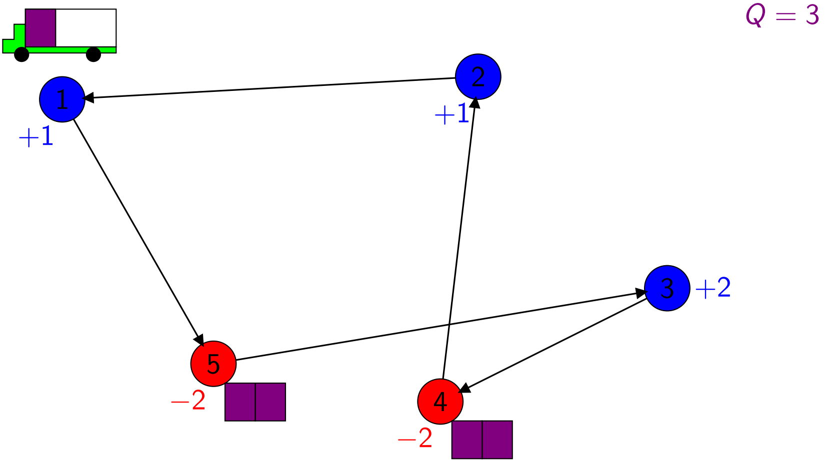
Remember: the depot can provide extra-load to the vehicle



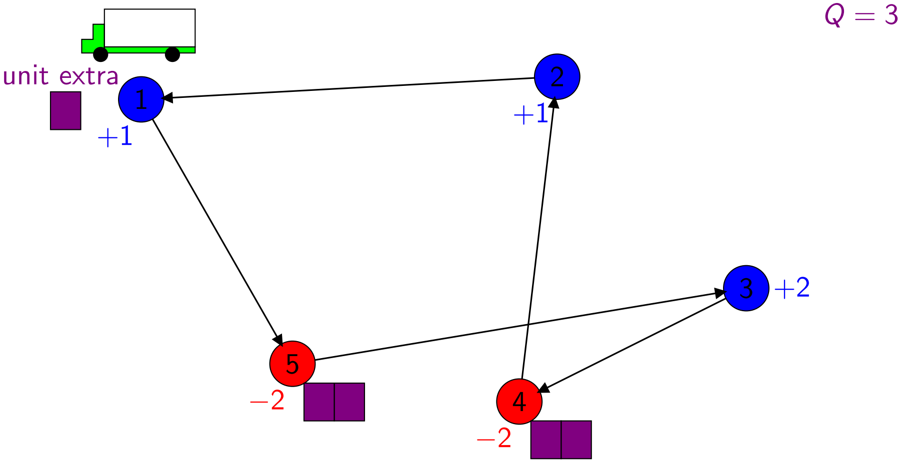
Remember: the depot can provide extra-load to the vehicle



Remember: the depot can provide extra-load to the vehicle



Remember: the depot can provide extra-load to the vehicle



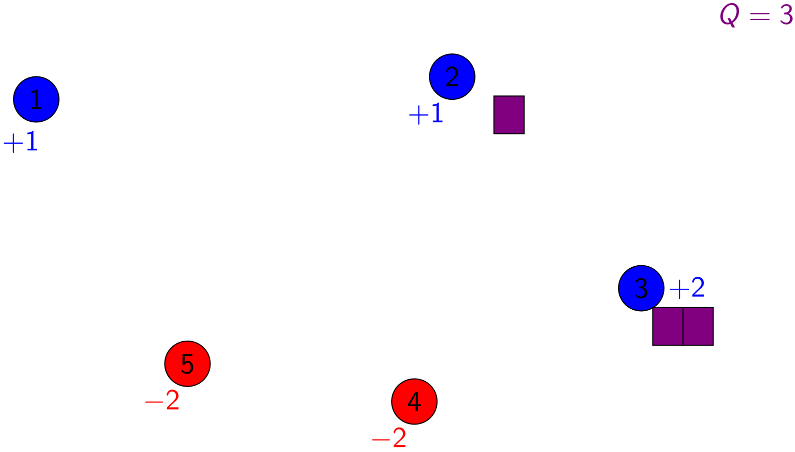
But the additional constraint of starting with a fix load from the depot can be imposed by splitting the depot into two dummy customers $1'$ and $1''$ where:

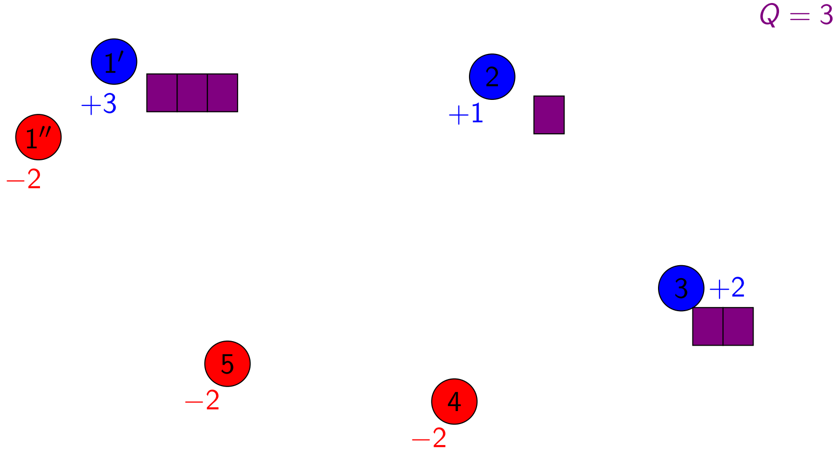
- $q_{1'} = +Q$ and $q_{1''} = q_1 - Q$ if $q_1 \geq 0$;
- $q_{1'} = +Q + q_1$ and $q_{1''} = -Q$ if $q_1 < 0$.

But the additional constraint of starting with a fix load from the depot can be imposed by splitting the depot into two dummy customers $1'$ and $1''$ where:

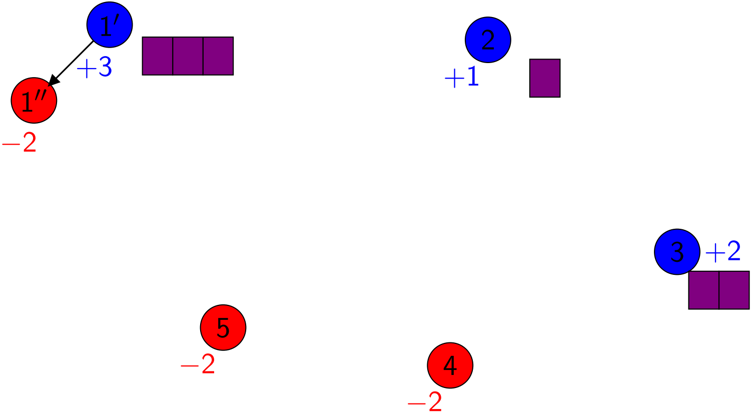
- $q_{1'} = +Q$ and $q_{1''} = q_1 - Q$ if $q_1 \geq 0$;
- $q_{1'} = +Q + q_1$ and $q_{1''} = -Q$ if $q_1 < 0$.

The arc from the pickup dummy vertex to the delivery dummy vertex must be also routed by the vehicle.


$Q = 3$
1
+1
2
+1
3
+2
5
−2
4
−2


$Q = 3$
$1'$
$+3$
$1''$
-2
2
$+1$
3
$+2$
5
-2
4
-2

$Q = 3$



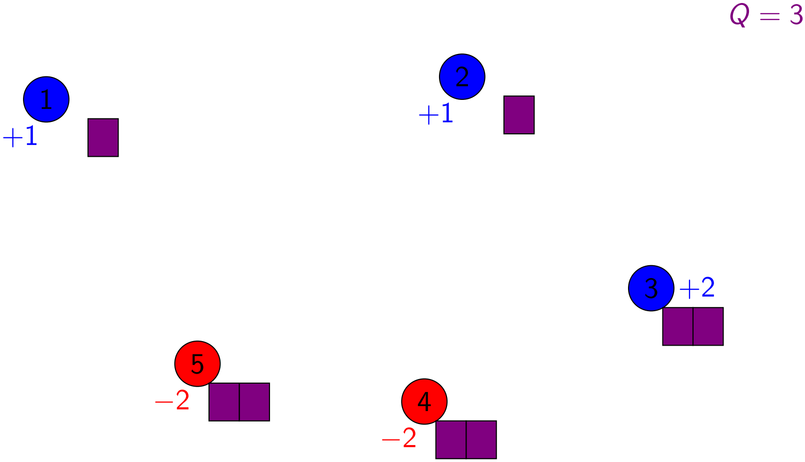
The special version of the 1-PDTSP where the delivery and pickup quantities are all equal to one has been studied:

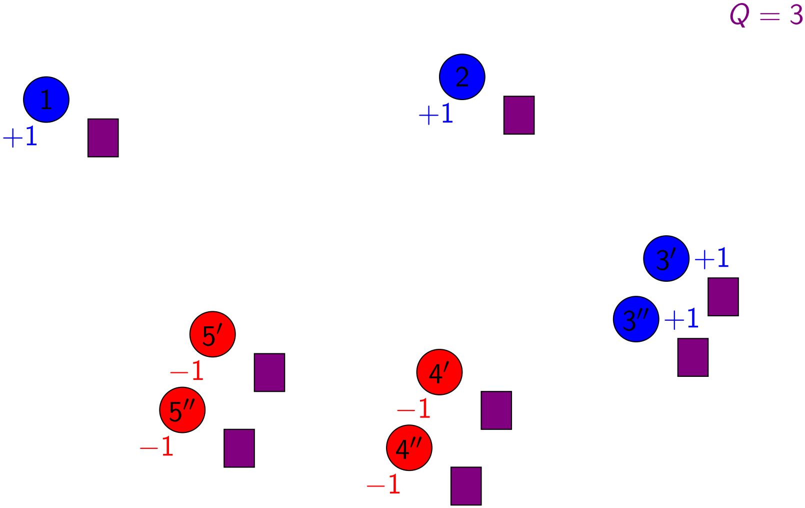
- Chalasani & Motwani (1999) call this problem *Q-delivery TSP* and propose heuristic algorithms.
- Anily & Bramel (1999) call this problem *Capacitated TSP with Pickups and Deliveries* (CTSPPD) and also propose heuristic algorithms.

The special version of the 1-PDTSP where the delivery and pickup quantities are all equal to one has been studied:

- Chalasani & Motwani (1999) call this problem *Q-delivery TSP* and propose heuristic algorithms.
- Anily & Bramel (1999) call this problem *Capacitated TSP with Pickups and Deliveries* (CTSPPD) and also propose heuristic algorithms.

If the Hamiltonian requirement on the route is relaxed, a 1-PDTSP instance can be solved as a CTSPPD instance splitting each customer i in q_i dummy customers.


$Q = 3$
1
+1
2
+1
3
+2
5
−2
4
−2


$Q = 3$
1
+1
2
+1
3′
+1
3″
+1
5′
−1
5″
−1
4′
−1
4″
−1

The *TSP with Pickups and Deliveries* (TSPPD).

- Also called *TSP with Pickups and Backhauls* (TSPPB) and *TSP with Delivery and Collection constraints* (TSPDC).

The *TSP with Pickups and Deliveries* (TSPPD).

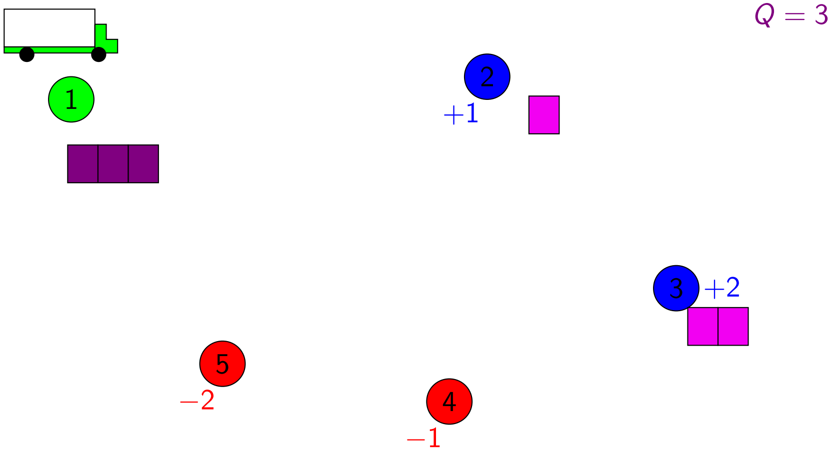
- Also called *TSP with Pickups and Backhauls* (TSPPB) and *TSP with Delivery and Collection constraints* (TSPDC).
- There are two types of products (i.e., two commodities).
  - The total amount of product collected from pickup customers must be delivered only at the depot (i.e., many-to-one).
  - The product collected from the depot must be delivered to the delivery customers (i.e., one-to-many).

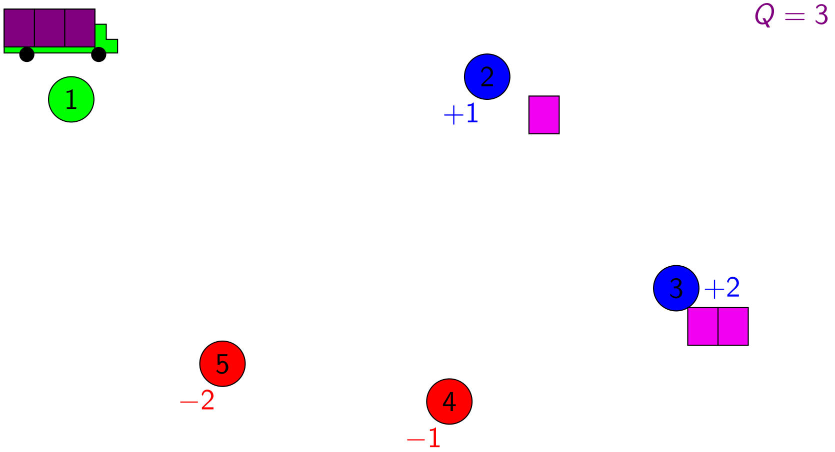
The *TSP with Pickups and Deliveries* (TSPPD).

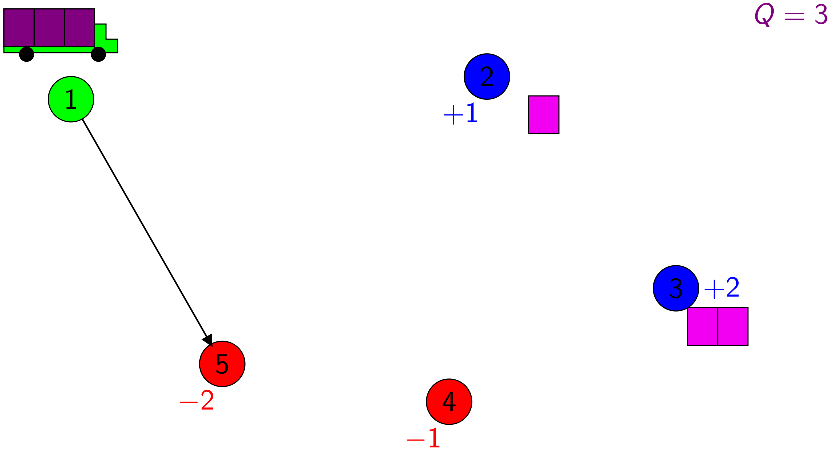
- Also called *TSP with Pickups and Backhauls* (TSPPB) and *TSP with Delivery and Collection constraints* (TSPDC).
- There are two types of products (i.e., two commodities).
  - The total amount of product collected from pickup customers must be delivered only at the depot (i.e., many-to-one).
  - The product collected from the depot must be delivered to the delivery customers (i.e., one-to-many).
- Bibliography:
  - MOSHEIOV (1994) introduces the TSPPD and proposes applications and heuristic approaches.
  - ANILY & MOSHEIOV (1994), and GENDREAU, LAPORTE & VIGO (1999) present approximation algorithms.
  - BALDACCI, HADJICONSTANTINOU & MINGOZZI (2003) present an exact algorithm.

The *TSP with Pickups and Deliveries* (TSPPD).

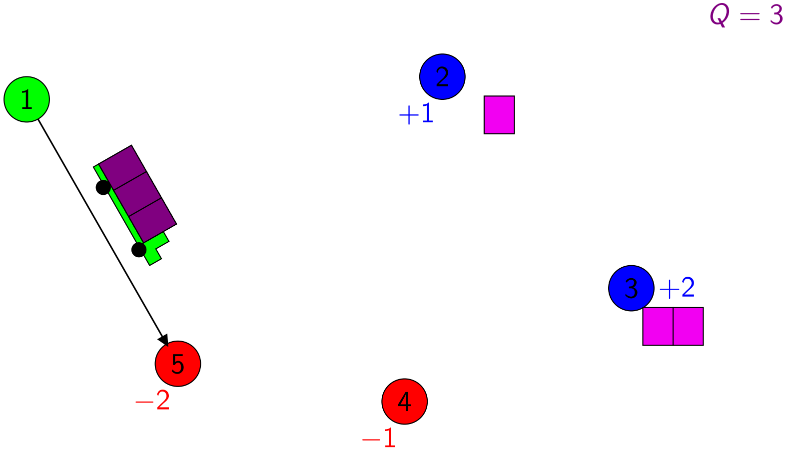
- Also called *TSP with Pickups and Backhauls* (TSPPB) and *TSP with Delivery and Collection constraints* (TSPDC).
- There are two types of products (i.e., two commodities).
  - The total amount of product collected from pickup customers must be delivered only at the depot (i.e., many-to-one).
  - The product collected from the depot must be delivered to the delivery customers (i.e., one-to-many).
- Bibliography:
  - MOSHEIOV (1994) introduces the TSPPD and proposes applications and heuristic approaches.
  - ANILY & MOSHEIOV (1994), and GENDREAU, LAPORTE & VIGO (1999) present approximation algorithms.
  - BALDACCI, HADJICONSTANTINOU & MINGOZZI (2003) present an exact algorithm.
- Example (empty bottles and full bottles).


$Q = 3$
2
+1
1
3
+2
5
−2
4
−1

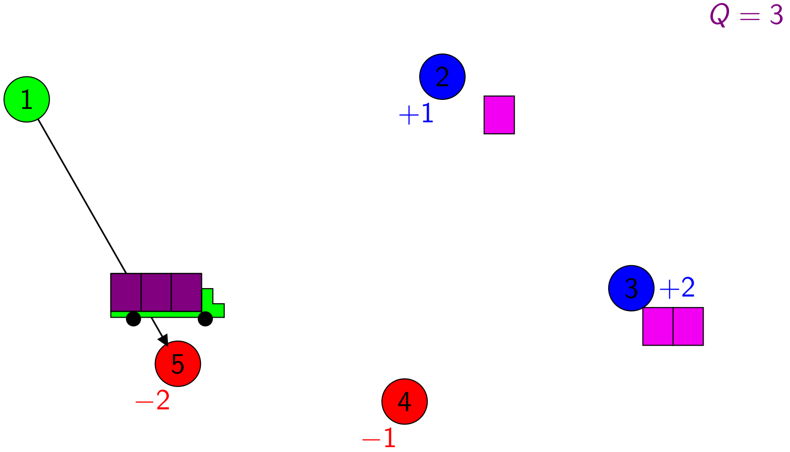

$Q = 3$
1
2
+1
3
+2
4
−1
5
−2



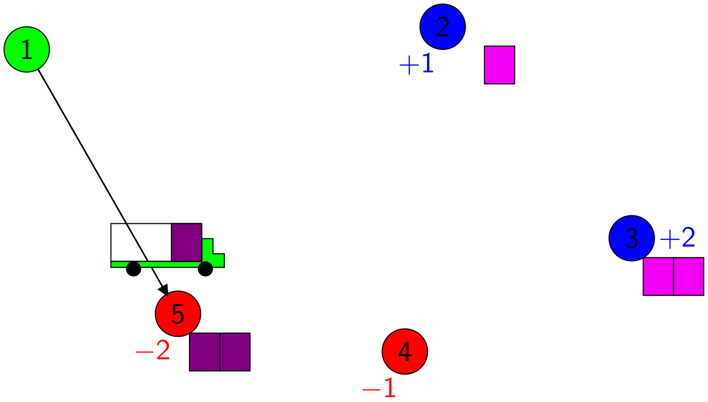
$Q = 3$
1
2
+1
3
+2
4
−1
5
−2

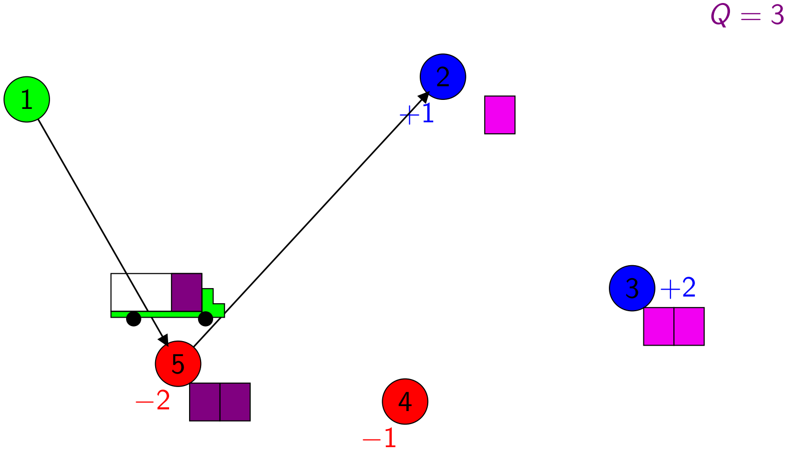


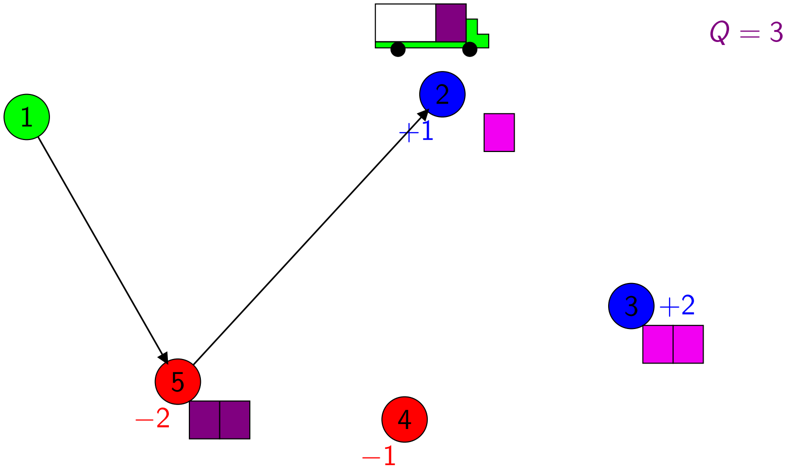
$Q = 3$
1
2
+1
3
+2
4
−1
5
−2

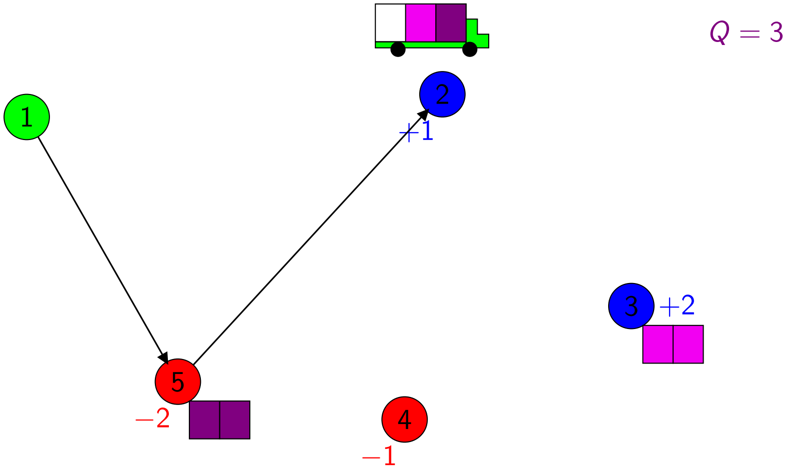

$Q = 3$
1
2
+1
3
+2
4
−1
5
−2

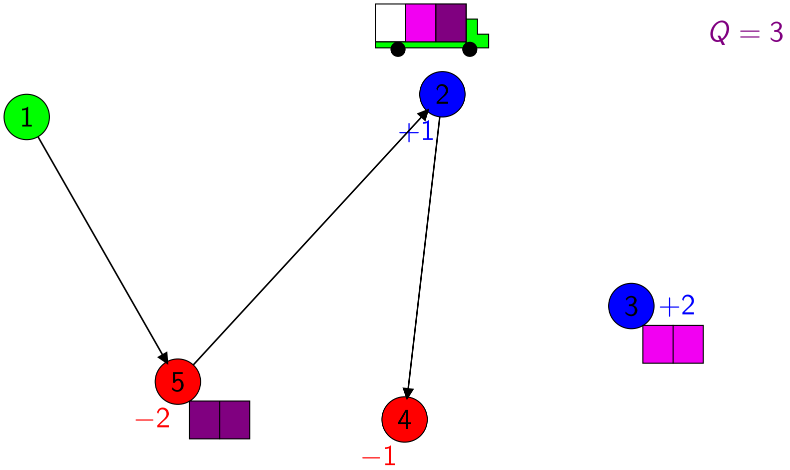
$Q = 3$

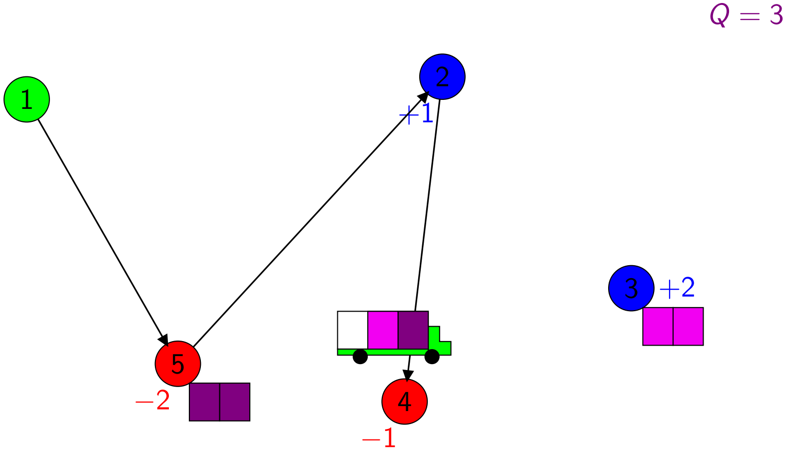



$Q = 3$
1
2
+1
3
+2
4
−1
5
−2

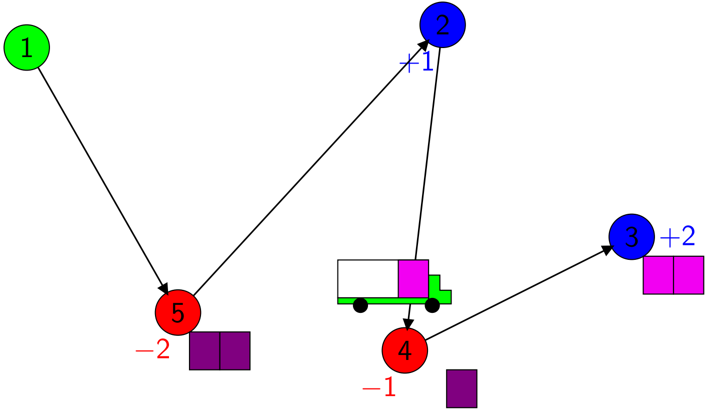

$Q = 3$
1
2
+1
3
+2
4
−1
5
−2

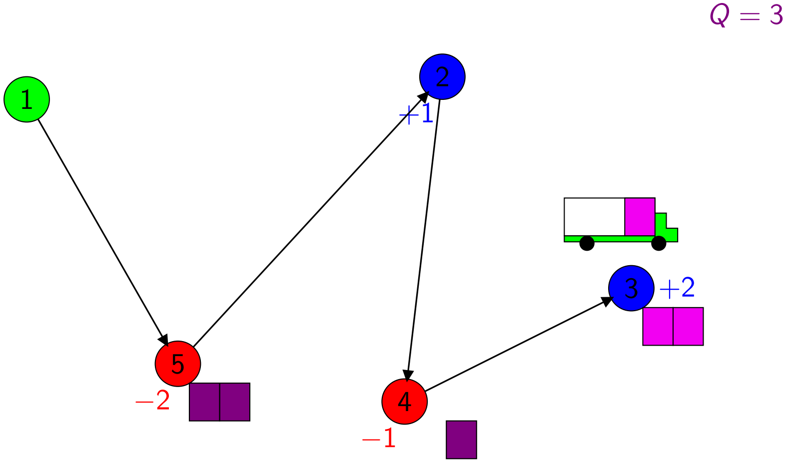

$Q = 3$
1
2
+1
3
+2
4
−1
5
−2

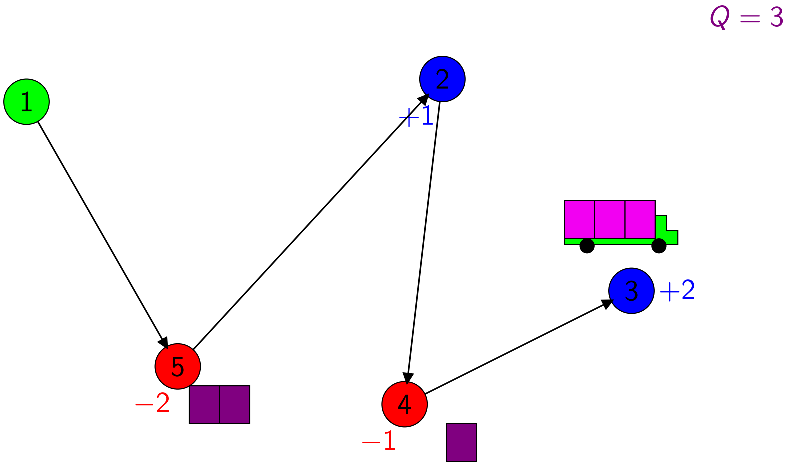

$Q = 3$
1
2
+1
3
+2
4
−1
5
−2

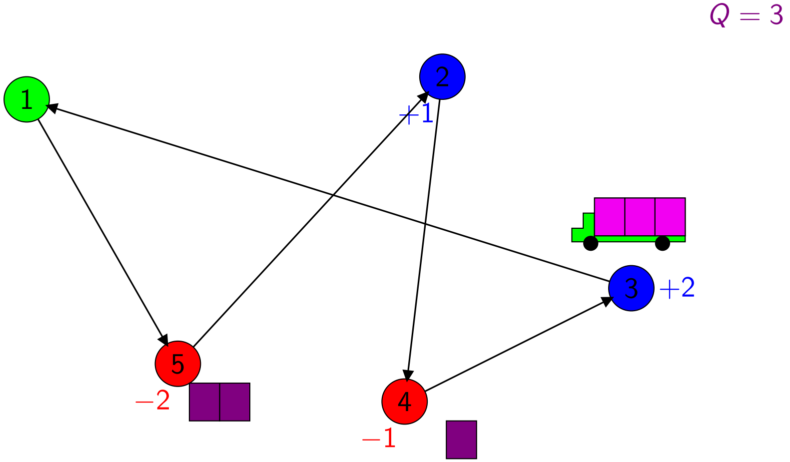

$Q = 3$
1
2
+1
3
+2
5
−2
4
−1

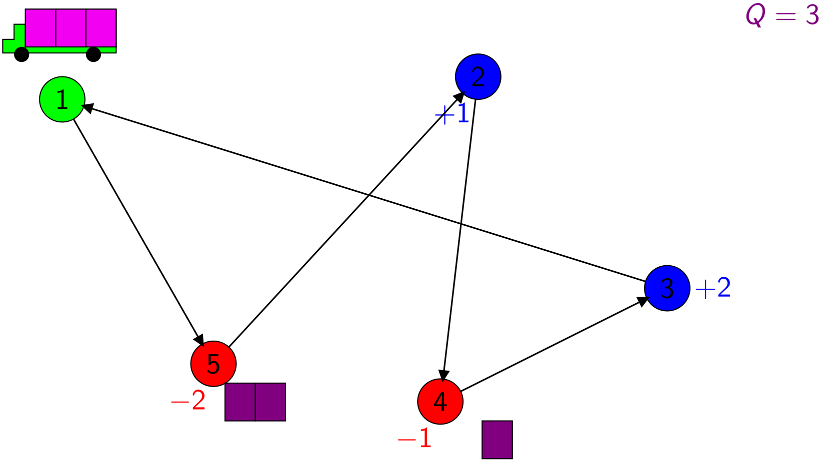
$Q = 3$




$Q = 3$
1
2
+1
3
+2
4
−1
5
−2

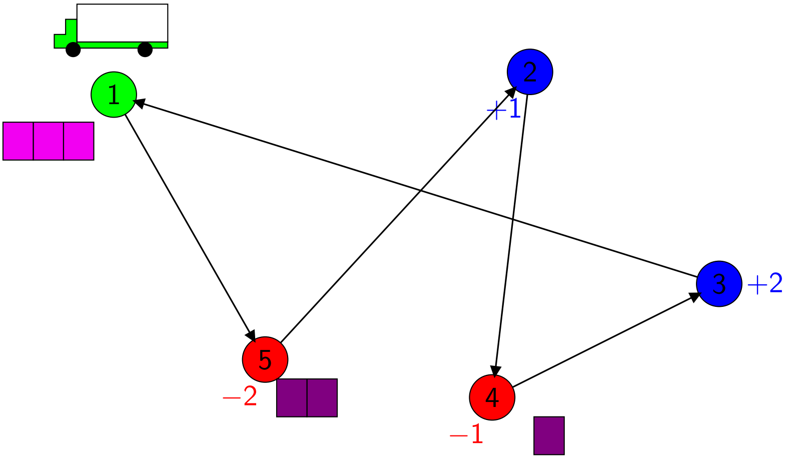

$Q = 3$
1
2
+1
3
+2
4
−1
5
−2


$Q = 3$
1
2
+1
3
+2
4
−1
5
−2



$Q = 3$
1
2
+1
3
+2
4
−1
5
−2

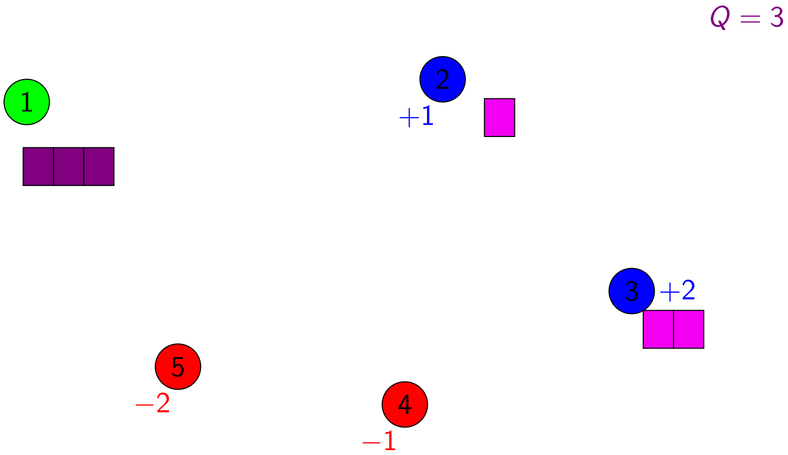
$Q = 3$

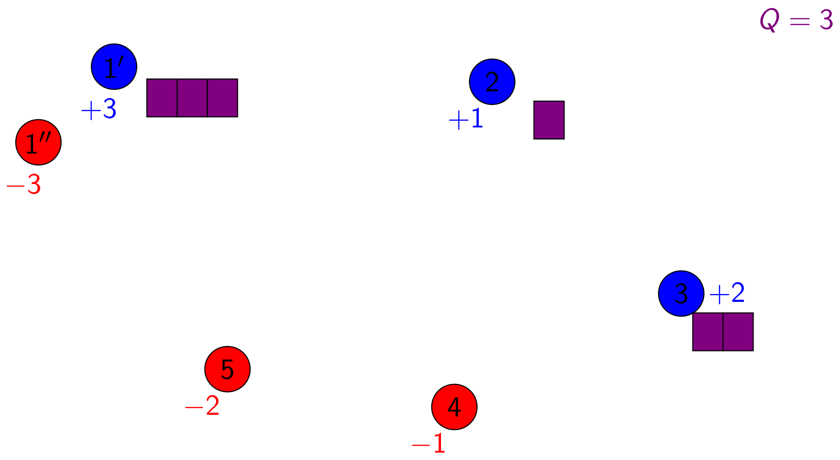


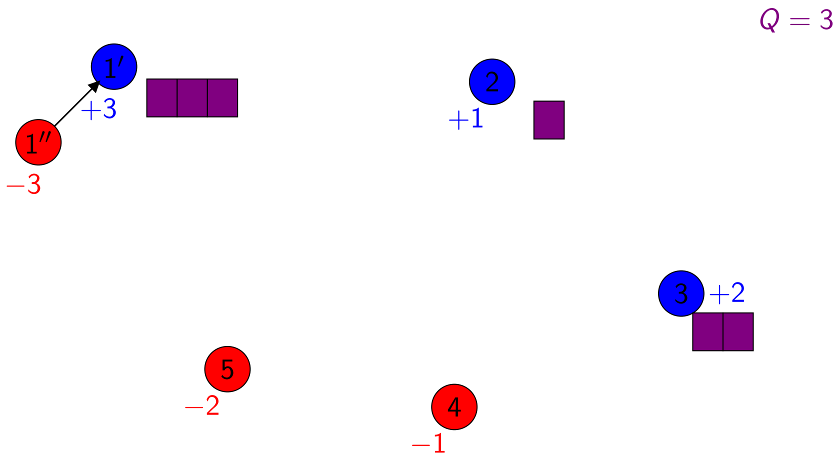
Each instance of the TSPPD can be transformed in an 1-PDTSP instance:

Each instance of the TSPPD can be transformed in an 1-PDTSP instance:

- splitting the depot in two different customers $1'$ and $1''$,
- fixing the arc variable between these customers.


$Q = 3$
1
2
$+1$
3
$+2$
4
-1
5
-2


$Q = 3$
$1'$
$+3$
$1''$
-3
2
$+1$
3
$+2$
5
-2
4
-1


$Q = 3$
$1'$
$+3$
$1''$
-3
2
$+1$
3
$+2$
5
-2
4
-1

This table summarizes some single-vehicle problems with pickup and delivery products:

Problem name	#	Origins-destinations	Hamilt.	Preemption	Q	Load
Swapping Problem	m	many-to-many	no	yes	1	yes
Stacker Crane Problem	m	one-to-one	no	no	1	yes
CDARP	m	one-to-one	no	no	k	yes
PDTSP	m	one-to-one	yes	no	∞	yes
TSPPD, TSPDB, TSPDC	2	one-to-many	yes	no	k	yes
TSPB	2	one-to-many	yes	no	∞	yes
CTSPPD, Q-delivery TSP	1	many-to-many	no	no	k	yes
1-PDTSP	1	many-to-many	yes	no	k	no

This talk concerns an algorithm for the 1-PDTSP, which contains the TSPPD.

- $V := \{1, ..., n\}$ is the set of vertices and E is the set of the edges (undirected model).

- $V := \{1, ..., n\}$ is the set of vertices and E is the set of the edges (undirected model).
- The edge $e \in E$ between i and j is denoted by $[i, j]$.

- $V := \{1, ..., n\}$ is the set of vertices and E is the set of the edges (undirected model).
- The edge $e \in E$ between i and j is denoted by $[i, j]$.
- For each $S, T \subset V$ we denote:
  - $\delta(S) := \{[i,j] \in E : i \in S,\ j \in V \setminus S\}$,
  - $E(S) := \{[i,j] \in E : i,j \in S\}$,
  - $E(S : T) := \{[i,j] \in E : i \in S,\ j \in T\}$.

- $V := \{1, \ldots, n\}$ is the set of vertices and E is the set of the edges (undirected model).
- The edge $e \in E$ between i and j is denoted by $[i, j]$.
- For each $S, T \subset V$ we denote:
  - $\delta(S) := \{[i,j] \in E : i \in S,\ j \in V \setminus S\}$,
  - $E(S) := \{[i,j] \in E : i,j \in S\}$,
  - $E(S : T) := \{[i,j] \in E : i \in S,\ j \in T\}$.
- We assume $q_1 = -\sum_{i=2}^{n} q_i$.

- $V := \{1, ..., n\}$ is the set of vertices and E is the set of the edges (undirected model).
- The edge $e \in E$ between i and j is denoted by $[i, j]$.
- For each $S, T \subset V$ we denote:
  - $\delta(S) := \{[i,j] \in E : i \in S,\ j \in V \setminus S\}$,
  - $E(S) := \{[i,j] \in E : i,j \in S\}$,
  - $E(S : T) := \{[i,j] \in E : i \in S,\ j \in T\}$.
- We assume $q_1 = -\sum_{i=2}^{n} q_i$.
- c_{ij} is the cost of going from i to j.

- $V := \{1, ..., n\}$ is the set of vertices and E is the set of the edges (undirected model).
- The edge $e \in E$ between i and j is denoted by $[i, j]$.
- For each $S, T \subset V$ we denote:
  - $\delta(S) := \{[i,j] \in E : i \in S,\ j \in V \setminus S\}$,
  - $E(S) := \{[i,j] \in E : i,j \in S\}$,
  - $E(S : T) := \{[i,j] \in E : i \in S,\ j \in T\}$.
- We assume $q_1 = -\sum_{i=2}^{n} q_i$.
- c_{ij} is the cost of going from i to j.
- $x_e := \begin{cases} 1 & \text{iff e is routed,} \\ 0 & \text{otherwise.} \end{cases}$

- $V := \{1, ..., n\}$ is the set of vertices and E is the set of the edges (undirected model).
- The edge $e \in E$ between i and j is denoted by $[i, j]$.
- For each $S, T \subset V$ we denote:
  - $\delta(S) := \{[i,j] \in E : i \in S,\ j \in V \setminus S\}$,
  - $E(S) := \{[i,j] \in E : i,j \in S\}$,
  - $E(S : T) := \{[i,j] \in E : i \in S,\ j \in T\}$.
- We assume $q_1 = -\sum_{i=2}^{n} q_i$.
- c_{ij} is the cost of going from i to j.
- $x_e := \begin{cases} 1 & \text{iff e is routed,} \\ 0 & \text{otherwise.} \end{cases}$
- For each $E' \subset E$ we denote $x(E') := \sum_{e \in E'} x_e$

A 0-1 ILP model for (symmetric) 1-PDTSP is:

$$\min \sum_{e \in E} c_e x_e$$

subject to

$$\begin{aligned} x(\delta(\{i\})) &= 2 && \text{for all } i \in V \\ x(\delta(S)) &\geq 2 && \text{for all } S \subset V \\ x(\delta(S)) &\geq \frac{2}{Q}\left|\sum_{i \in S} q_i\right| && \text{for all } S \subset V \\ x_e &\in \{0,1\} && \text{for all } e \in E. \end{aligned}$$

This model is obtained by Benders' decomposition over the continuous variables (flow) of a mixed model.

- Clearly, the 1-PDTSP is an $\mathcal{NP}$-hard optimization problem in the strong sense since it coincides with TSP when the vehicle capacity is large enough.

- Clearly, the 1-PDTSP is an $\mathcal{NP}$-hard optimization problem in the strong sense since it coincides with TSP when the vehicle capacity is large enough.
- Even more, checking if there is a feasible solution of a 1-PDTSP instance is $\mathcal{NP}$-complete in the strong sense. The idea is because 3-Partition Problem is a particular case of the 1-PDTSP.

Inequalities derived from the TSP are:

- *2-matching inequalities*.
- *Comb inequalities* (see LETCHFORD & LODI (2002) for some recent separation procedures).
- Etc., (see NADDEF (2001) for other valid inequalities).

The inequalities derived from the Benders' decomposition can be strengthened, i.e.:

$$x(\delta(S)) \geq 2r(S) \quad \text{for all } S \subset V,$$

where

$$r(S) := \max\left\{1, \left\lceil \left| \sum_{i \in S} q_i \right| / Q \right\rceil \right\}$$

is a lower bound of the number of times that the vehicle has to go inside/outside the customer set S.

The inequalities derived from the Benders' decomposition can be strengthened, i.e.:

$$x(\delta(S)) \geq 2r(S) \quad \text{for all } S \subset V,$$

where

$$r(S) := \max\left\{1, \left\lceil \left|\sum_{i \in S} q_i\right| / Q \right\rceil\right\}$$

is a lower bound of the number of times that the vehicle has to go inside/outside the customer set S.

- They are similar inequalities to the known *capacity constraints* for the CVRP.

The inequalities derived from the Benders' decomposition can be strengthened, i.e.:

$$x(\delta(S)) \geq 2r(S) \quad \text{for all } S \subset V,$$

where

$$r(S) := \max\left\{1, \left\lceil \left|\sum_{i \in S} q_i\right| / Q \right\rceil\right\}$$

is a lower bound of the number of times that the vehicle has to go inside/outside the customer set S.

- They are similar inequalities to the known *capacity constraints* for the CVRP.
- The *rounded Benders' cuts* can be rewritten as: $x(E(S)) \leq |S| - r(S) \quad \text{for all } S \subset V.$

Let $W_1, \ldots, W_m$ be subsets of V such that:

$$
\begin{aligned}
W_i \cap W_j &= \{v\} && \text{for all } 1 \leq i < j \leq m, \\
r(W_i) &= 1 && \text{for all } i \in \{1, \ldots, m\}, \\
r(W_i \cup W_j) &> 1 && \text{for all } 1 \leq i < j \leq m,
\end{aligned}
$$

a *clique cluster inequality* is:

$$\sum_{i=1}^{m} x(E(W_i)) \leq \sum_{i=1}^{m} |W_i| - 2m + 1.$$

This family of inequalities was first proposed by AUGERAT (1995) and POCHET (1998) for the CVRP.

Recent articles deal with the multistar inequalities for the CVRP with general demands (see, e.g., Blasum & Hochtättler (2000), Fisher (1995), Gouveia (1995) and Letchford, Eglese & Lysgaar (2003)).

Recent articles deal with the multistar inequalities for the CVRP with general demands (see, e.g., Blasum & Hochtättler (2000), Fisher (1995), Gouveia (1995) and Letchford, Eglese & Lysgaar (2003)).
They can be easily adapted for the 1-PDTSP:

Recent articles deal with the multistar inequalities for the CVRP with general demands (see, e.g., BLASUM & HOCHTÄTTLER (2000), FISHER (1995), GOUVEIA (1995) and LETCHFORD, EGLESE & LYSGAAR (2003)).
They can be easily adapted for the 1-PDTSP:

$$x(E(N)) \leq |N| - \frac{1}{Q}\left|\sum_{i \in N}\left(q_i + \sum_{j \in S} q_j x_{[i,j]}\right)\right|$$

for $N \subset V$ and $S \subset V \setminus N$.

- We call these inequalities *inhomogeneous multistar* inequalities (also, *general large multistar* inequalities).

Recent articles deal with the multistar inequalities for the CVRP with general demands (see, e.g., Blasum & Hochtättler (2000), Fisher (1995), Gouveia (1995) and Letchford, Eglese & Lysgaar (2003)).
They can be easily adapted for the 1-PDTSP:

$$x(E(N)) \leq |N| - \frac{1}{Q}\left|\sum_{i\in N}\left(q_i + \sum_{j\in S} q_j x_{[i,j]}\right)\right|$$

for $N \subset V$ and $S \subset V \setminus N$.

- We call these inequalities *inhomogeneous multistar* inequalities (also, *general large multistar* inequalities).
- Blasum & Hochtättler (2000) propose a polynomial separation procedure and give a *generalization* when each vertex in S is replaced by a subset of vertices for the CVRP.

Recent articles deal with the multistar inequalities for the CVRP with general demands (see, e.g., Blasum & Hochtättler (2000), Fisher (1995), Gouveia (1995) and Letchford, Eglese & Lysgaar (2003)).
They can be easily adapted for the 1-PDTSP:

$$x(E(N)) \leq |N| - \frac{1}{Q}\left|\sum_{i \in N}\left(q_i + \sum_{j \in S} q_j x_{[i,j]}\right)\right|$$

for $N \subset V$ and $S \subset V \setminus N$.

- We call these inequalities *inhomogeneous multistar* inequalities (also, *general large multistar* inequalities).
- Blasum & Hochtättler (2000) propose a polynomial separation procedure and give a *generalization* when each vertex in S is replaced by a subset of vertices for the CVRP.
- We call the generalized inequalities *generalized inhomogeneous multistar* inequalities.

Another kind of multistar inequalities is proposed by LETCHFORD, EGLESE & LYSGAAR (2003) for the CVRP.

Another kind of multistar inequalities is proposed by LETCHFORD, EGLESE & LYSGAAR (2003) for the CVRP. Let us consider the following subsets:

$$\begin{aligned} N &\subset V & \quad & \textit{Nucleus} \\ C &\subset N & & \textit{Collector} \\ S &\subset V \setminus N & & \textit{Satellite set} \end{aligned}$$

then

Another kind of multistar inequalities is proposed by LETCHFORD, EGLESE & LYSGAAR (2003) for the CVRP. Let us consider the following subsets:

$N \subset V$	*Nucleus*
$C \subset N$	*Collector*
$S \subset V \setminus N$	*Satellite set*

then

$$\lambda x(E(N)) + x(E(C:S)) \leq \mu$$

is valid for appropriate values of λ and μ.

Another kind of multistar inequalities is proposed by Letchford, Eglese & Lysgaar (2003) for the CVRP. Let us consider the following subsets:

$$\begin{array}{rl} N \subset V & \textit{Nucleus} \\ C \subset N & \textit{Collector} \\ S \subset V \setminus N & \textit{Satellite set} \end{array}$$

then

$$\lambda x(E(N)) + x(E(C:S)) \leq \mu$$

is valid for appropriate values of λ and μ.
When $C = N$ they are called *homogeneous multistar* inequalities; otherwise they are called *homogeneous partial multistar* inequalities.

$$\lambda x(E(N)) + x(E(C : S)) \leq \mu$$

- For a fixed N, C and S, Letchford, Eglese & Lysgaar (2003) give a procedure to find appropriate value of λ and μ. This procedure is adapted for the 1-PDTSP.

$$\lambda x(E(N)) + x(E(C:S)) \leq \mu$$

- For a fixed N, C and S, Letchford, Eglese & Lysgaar (2003) give a procedure to find appropriate value of λ and μ. This procedure is adapted for the 1-PDTSP.
- The homogeneous multistar (partial) inequalities can also be *generalized* when the vertices in S are replaced by a collection of subsets $\{S_1, \ldots, S_m\}$.

$$\lambda x(E(N)) + \sum_{i=1}^{m} \left(x(E(C:S_i)) + x(E(S_i)) \right) \leq \mu + \sum_{i=1}^{m} (|S_i| - 1).$$

The algorithm starts by solving

$$\min \sum_{e \in E} c_e x_e$$

subject to

$$\begin{aligned} x(\delta(\{i\})) &= 2 \qquad && \text{for all } i \in V \\ 0 \le x_e &\le 1 \qquad && \text{for all } e \in E \end{aligned}$$

After, other valid inequalities violated by the fractional solution are inserted dynamically.

Separation procedures for:

- Rounded Benders' cuts (*Bend.*).

$$x(E(S)) \leq |S| - r(S)$$

Separation procedures for:

- Rounded Benders' cuts (*Bend.*).
- Comb and other TSP inequalities are not implemented.

$$x(\delta(H)) + \sum_{i=1}^{m} x(\delta(T_i)) \geq 3m + 1, \ldots$$

Separation procedures for:

- Rounded Benders' cuts (*Bend.*).
- Comb and other TSP inequalities are not implemented.
- Clique clusters inequalities (*cliq.*).

$$\sum_{i=1}^{m} x(E(W_i)) \leq \sum_{i=1}^{m} |W_i| - 2m + 1$$

Separation procedures for:

- Rounded Benders' cuts (*Bend.*).
- Comb and other TSP inequalities are not implemented.
- Clique clusters inequalities (*cliq.*).
- Generalized homogeneous multistar inequalities (*ghm*).
- Generalized homogeneous partial multistar inequalities (*ghpm*).

$$\lambda x(E(N)) + \sum_{i=1}^{m} \big(x(E(C : S_i)) + x(E(S_i))\big) \leq \mu + \sum_{i=1}^{m}(|S_i| - 1)$$

Separation procedures for:

- Rounded Benders' cuts (*Bend.*).
- Comb and other TSP inequalities are not implemented.
- Clique clusters inequalities (*cliq.*).
- Generalized homogeneous multistar inequalities (*ghm*).
- Generalized homogeneous partial multistar inequalities (*ghpm*).
- Generalized inhomogeneous multistar inequalities (*gim*).

$$x(\delta(N)) \geq \frac{2}{Q}\left|\sum_{i\in N} q_i + \sum_{j=1}^{m}\left(\sum_{i\in S_j} q_i\right)\left(x(E(N:S_j)) + 2 - x(\delta(S_j))\right)\right|$$

- The algorithm has been implemented in ANSI C, and ran on a personal computer AMD Athlon XP 2600+ (2.08 Ghz.).
- The software CPLEX 7.0 has been used as LP-solver in the Branch-and-Cut algorithm.
- The time limit was 7200 seconds.
- We tested the performance of the 1-PDTSP approach by using the generator of random Euclidean instances proposed by MOSHEIOV (1994) for the TSPPD.
  - The customers are in the rectangle $[-500, 500] \times [-500, 500]$.
  - The demands q_i are integer numbers in $[-10, 10]$.
  - The travel costs c_{ij} are the Euclidean distances.

- The algorithm has been implemented in ANSI C, and ran on a personal computer AMD Athlon XP 2600+ (2.08 Ghz.).
- The software CPLEX 7.0 has been used as LP-solver in the Branch-and-Cut algorithm.
- The time limit was 7200 seconds.
- We tested the performance of the 1-PDTSP approach by using the generator of random Euclidean instances proposed by MOSHEIOV (1994) for the TSPPD.
  - The customers are in the rectangle $[-500, 500] \times [-500, 500]$.
  - The demands q_i are integer numbers in $[-10, 10]$.
  - The travel costs c_{ij} are the Euclidean distances.
- We also tested the algorithm over the TSPPD instances used in GENDREAU, LAPORTE & VIGO (1999) and BALDACCI, HADJICONSTANTINOU & MINGOZZI (2003).

n	Q	Cuts *Bend.*	*cliq.*	*ghm*	*ghpm*	*gim*	LB/z^*	z^*/TSP	B&C	Time	t.l.
60	TSPPD	63.4	0.0	10.5	0.6	0.0	98.51	101.33	102.9	1.06	0
60	10	705.6	147.1	590.0	65.2	22.1	96.60	144.86	2650.7	210.45	0
60	15	407.5	34.9	299.8	24.4	2.6	97.81	119.82	739.2	50.71	0
60	20	223.4	6.9	135.9	14.2	2.0	98.38	109.88	109.4	5.94	0
60	25	156.4	1.4	87.6	5.4	0.4	98.47	105.60	84.1	2.38	0
60	30	118.4	0.7	42.1	2.3	0.2	98.73	103.00	140.9	2.35	0
60	100	35.4	0.0	0.0	0.0	0.0	98.65	100.00	42.5	0.46	0
80	TSPPD	90.8	0.1	49.1	0.4	0.0	98.81	101.06	72.3	2.28	0
80	10	1030.4	166.0	804.6	67.6	68.2	97.12	153.72	4615.0	704.26	5
80	15	1418.3	125.3	605.9	51.1	12.7	96.88	126.83	6151.7	989.76	3
80	20	1590.0	112.5	356.1	32.8	6.6	96.85	117.52	18357.1	2079.17	2
80	25	800.8	14.7	309.9	33.6	3.0	97.41	109.68	2646.4	193.74	1
80	30	465.5	4.8	139.2	16.0	0.5	98.16	106.34	1491.2	82.64	0
80	100	44.6	0.0	0.0	0.0	0.0	99.17	100.00	146.2	1.27	0
100	TSPPD	148.3	0.1	29.2	0.5	0.0	98.72	101.01	261.3	6.69	0
100	20	1684.3	73.0	435.3	27.7	1.0	97.22	110.79	18674.0	2733.24	7
100	25	1841.1	23.8	365.0	29.3	1.3	97.34	109.83	9404.0	1758.84	2
100	30	1404.0	7.7	248.4	21.5	0.9	97.74	106.84	11728.7	1401.44	0
100	100	54.3	0.0	0.0	0.0	0.0	99.22	100.00	42.6	1.45	0

Table: Average results of the random Euclidean instances

β	LB/Opt.	Opt/TSP	B&C	Time
0.00	99.67	100.00	1172.0	64.99
0.05	99.61	100.54	2888.5	187.37
0.10	99.54	100.66	816.1	48.10
0.20	99.49	100.85	1505.0	95.98

Table: Results of the TSPPD instances derived from VRP test problems described in GENDREAU, LAPORTE & VIGO (1999)

β	n	LB/Opt.	Opt./TSP	B&C	Time
0.05	50	99.36	100.17	4.8	0.09
0.05	100	99.15	100.26	219.3	3.52
0.05	150	99.15	100.13	703.7	23.44
0.05	200	99.15	100.34	2650.9	113.84
0.10	50	99.49	100.44	3.7	0.11
0.10	100	98.96	100.48	209.0	4.86
0.10	150	99.08	100.33	5279.4	158.98
0.10	200	99.11	100.43	1255.0	77.85
0.20	50	99.39	100.79	5.9	0.13
0.20	100	98.84	100.82	952.5	16.09
0.20	150	99.02	100.51	2812.7	101.16
0.20	200	99.02	100.59	5058.0	249.40
∞	50	98.12	102.42	506.9	1.97
∞	100	98.76	100.74	1646.2	18.33
∞	150	98.99	100.43	2108.7	57.32
∞	200	99.08	100.45	7967.5	513.01

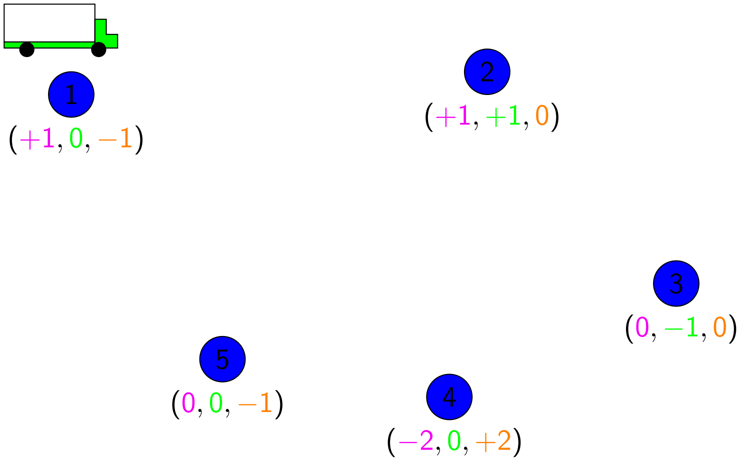
Table: Results of the Euclidian TSPPD instances described in Gendreau, Laporte & Vigo (1999)

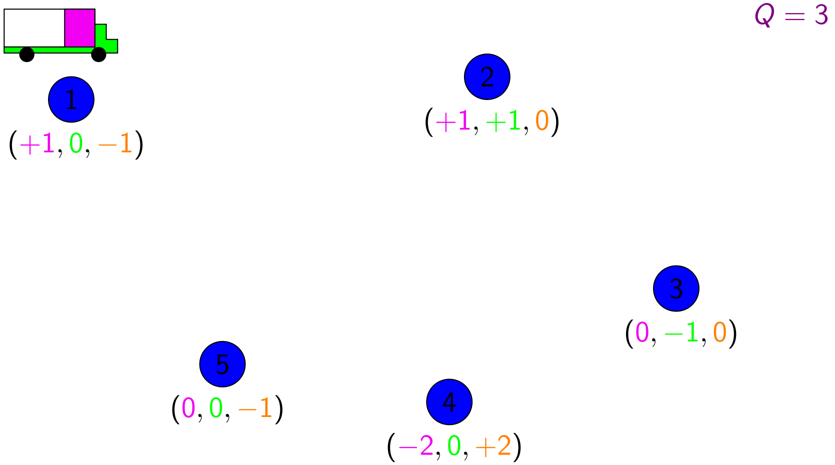
(We are obtaining better result than Baldacci et al. (2003))

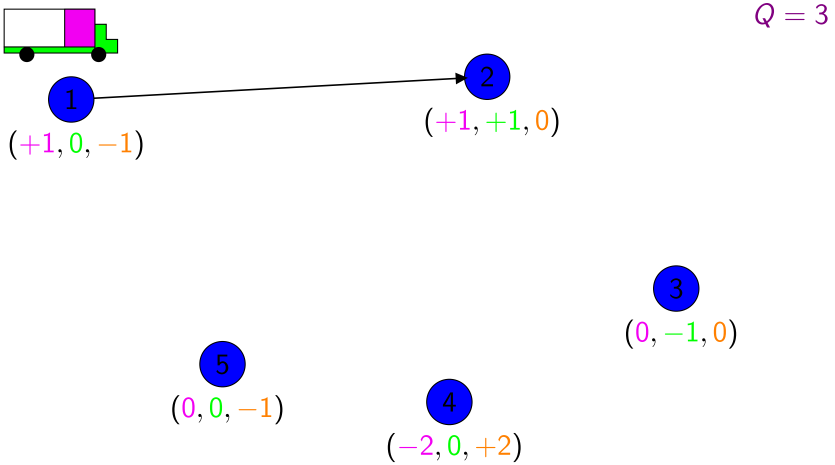
β	n	LB/Opt.	Opt/TSP	B&C	Time
0.05	50	99.72	100.20	1.4	0.06
0.05	100	99.82	100.64	7.7	0.81
0.05	150	99.83	100.53	3.4	1.78
0.05	200	99.77	100.07	3.5	3.51
0.10	50	99.45	100.97	2.7	0.09
0.10	100	99.75	100.86	4.8	0.82
0.10	150	99.77	100.60	9.4	2.99
0.10	200	99.70	100.23	4.9	4.42
0.20	50	99.43	101.08	5.0	0.13
0.20	100	99.66	100.98	26.2	2.23
0.20	150	99.66	100.79	28.4	7.20
0.20	200	99.70	100.46	14.5	8.60
∞	50	99.41	101.28	7.2	0.16
∞	100	99.69	100.63	16.1	1.52
∞	150	99.53	101.33	6.2	2.33
∞	200	99.40	100.52	17.8	10.38

Table: Results of the symmetric TSPPD instances described in GENDREAU, LAPORTE & VIGO (1999)

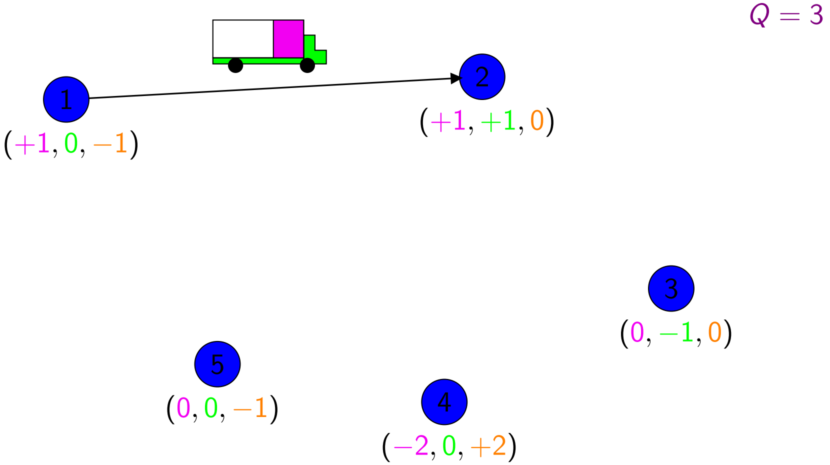
$Q = 3$

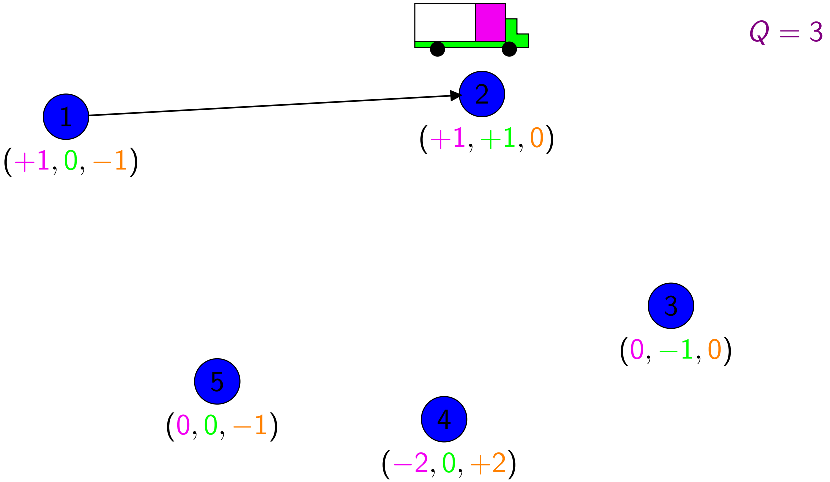


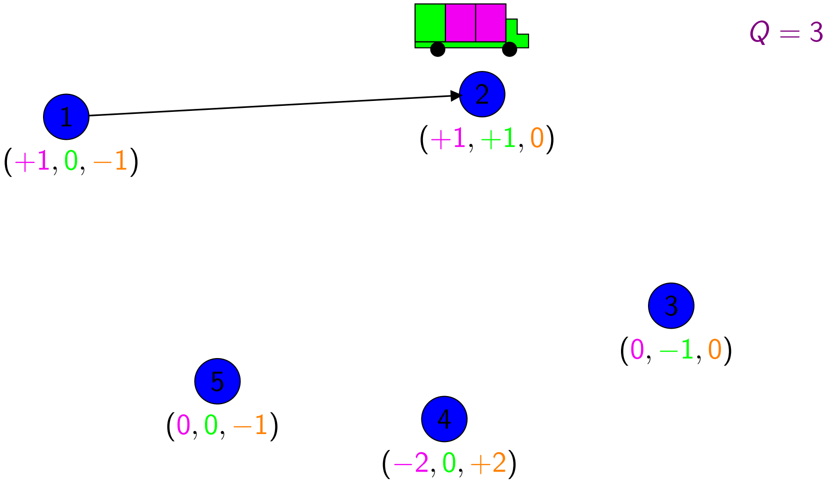

$Q = 3$
1
$(+1, 0, -1)$
2
$(+1, +1, 0)$
3
$(0, -1, 0)$
4
$(-2, 0, +2)$
5
$(0, 0, -1)$



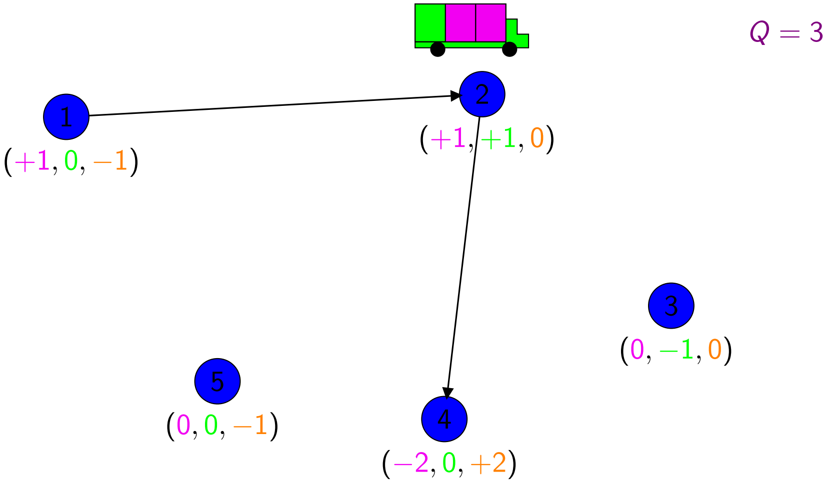
$Q = 3$
1
$(+1, 0, -1)$
2
$(+1, +1, 0)$
3
$(0, -1, 0)$
4
$(-2, 0, +2)$
5
$(0, 0, -1)$

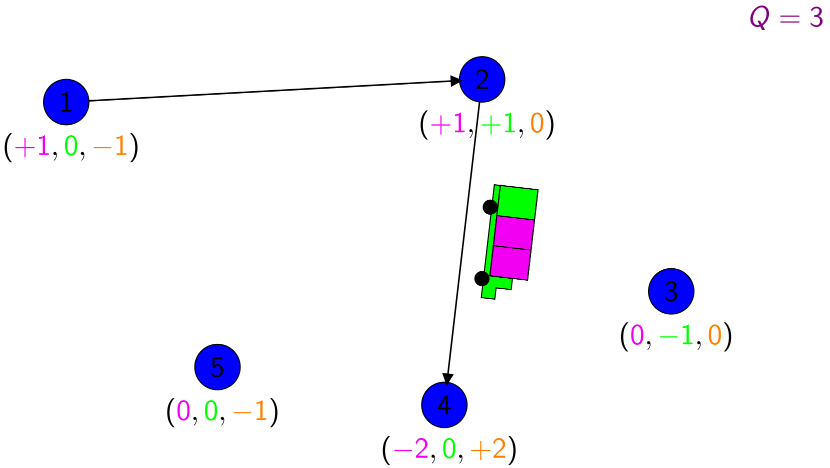

$Q = 3$
1
$(+1, 0, -1)$
2
$(+1, +1, 0)$
3
$(0, -1, 0)$
4
$(-2, 0, +2)$
5
$(0, 0, -1)$


$Q = 3$
1
$(+1, 0, -1)$
2
$(+1, +1, 0)$
3
$(0, -1, 0)$
4
$(-2, 0, +2)$
5
$(0, 0, -1)$

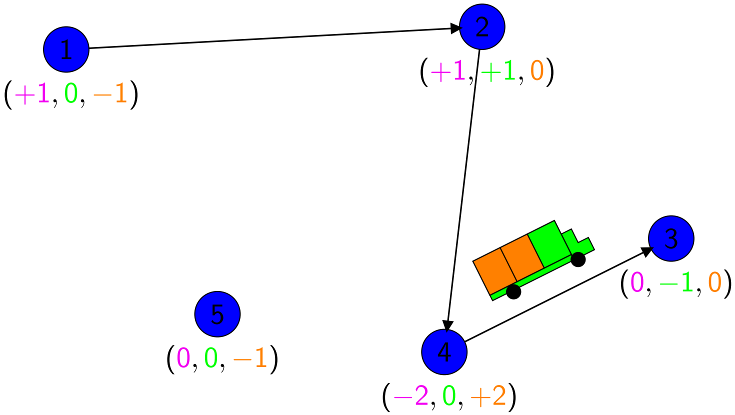


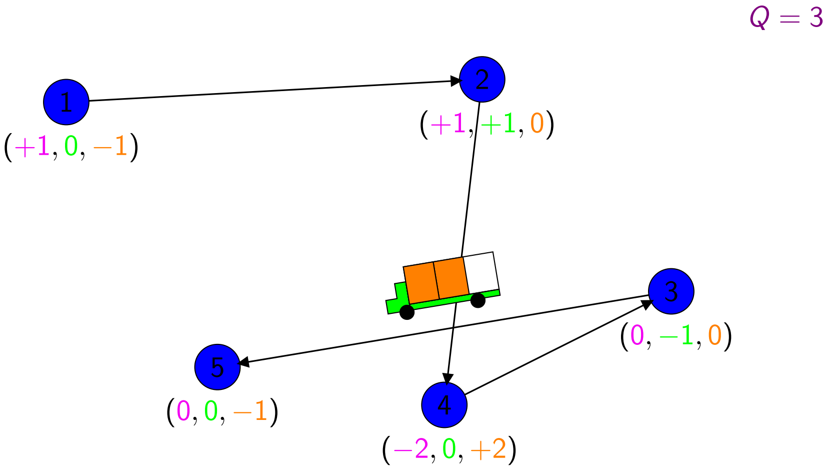
$Q = 3$
1
$(+1, 0, -1)$
2
$(+1, +1, 0)$
3
$(0, -1, 0)$
4
$(-2, 0, +2)$
5
$(0, 0, -1)$


$Q = 3$
1
$(+1, 0, -1)$
2
$(+1, +1, 0)$
3
$(0, -1, 0)$
4
$(-2, 0, +2)$
5
$(0, 0, -1)$


$Q = 3$
1
$(+1, 0, -1)$
2
$(+1, +1, 0)$
3
$(0, -1, 0)$
4
$(-2, 0, +2)$
5
$(0, 0, -1)$

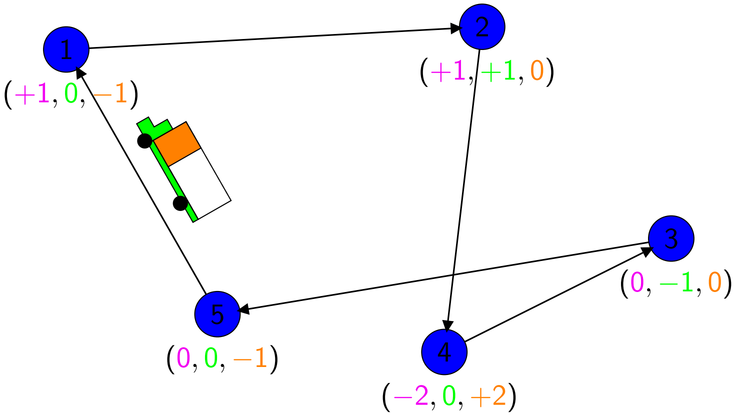
$Q = 3$

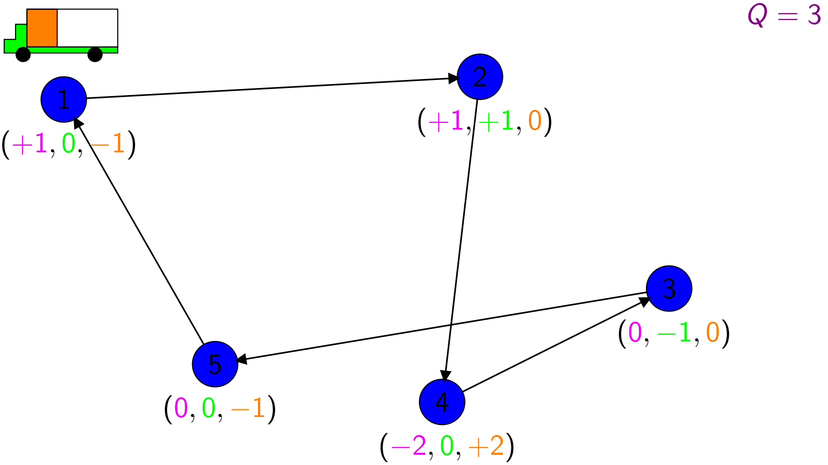




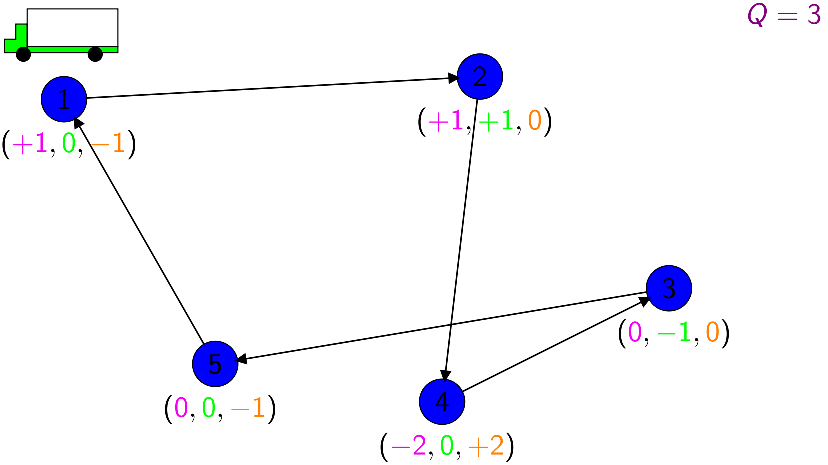
$Q = 3$
1
$(+1, 0, -1)$
2
$(+1, +1, 0)$
3
$(0, -1, 0)$
4
$(-2, 0, +2)$
5
$(0, 0, -1)$

$Q = 3$





$Q = 3$
1
$(+1, 0, -1)$
2
$(+1, +1, 0)$
3
$(0, -1, 0)$
4
$(-2, 0, +2)$
5
$(0, 0, -1)$


$Q = 3$
1
$(+1, 0, -1)$
2
$(+1, +1, 0)$
3
$(0, -1, 0)$
4
$(-2, 0, +2)$
5
$(0, 0, -1)$

Asymmetric Model for the m-PDTSP:

$$\min \sum_{a \in A} c_a x_a$$

subject to

$$\begin{aligned}
x(\delta^-(\{i\})) &= 1 && \text{for all } i \in V \\
x(\delta^+(\{i\})) &= 1 && \text{for all } i \in V \\
x(\delta^+(S)) &\geq 1 && \text{for all } S \subset V \\
x_a &\in \{0,1\} && \text{for all } a \in A \\
\sum_{a \in \delta^+(\{i\})} f_a^k - \sum_{a \in \delta^-(\{i\})} f_a^k &= q_i && \text{for all } i \in V, k = 1, \ldots, m \\
0 \leq \sum_{k=1}^{m} f_a^k &\leq Q x_a && \text{for all } a \in A.
\end{aligned}$$

Preliminary observations:

- The Stacker Crane Problem, CDARP and PDTSP are particular case of the m-PDTSP (also the TSPPD, CTSPPD and 1-PDTSP).

Preliminary observations:

- The Stacker Crane Problem, CDARP and PDTSP are particular case of the m-PDTSP (also the TSPPD, CTSPPD and 1-PDTSP).
- Some instances of 20 customers cannot be solved (using CPLEX as black box).

Preliminary observations:

- The Stacker Crane Problem, CDARP and PDTSP are particular case of the m-PDTSP (also the TSPPD, CTSPPD and 1-PDTSP).
- Some instances of 20 customers cannot be solved (using CPLEX as black box).
- A tour can be feasible in one direction and non-feasible in the other direction.

Preliminary observations:

- The Stacker Crane Problem, CDARP and PDTSP are particular case of the m-PDTSP (also the TSPPD, CTSPPD and 1-PDTSP).
- Some instances of 20 customers cannot be solved (using CPLEX as black box).
- A tour can be feasible in one direction and non-feasible in the other direction.

Future research.

- Heuristics procedures.

Preliminary observations:

- The Stacker Crane Problem, CDARP and PDTSP are particular case of the m-PDTSP (also the TSPPD, CTSPPD and 1-PDTSP).
- Some instances of 20 customers cannot be solved (using CPLEX as black box).
- A tour can be feasible in one direction and non-feasible in the other direction.

Future research.

- Heuristics procedures.
- Separation procedures for and branch-and-cut algorithm.

That's all