

**Ricollocazione nel bike-sharing**

**Matteo Gianvenuti**

**Donato Bruno**

Sommario

[Problema 3](#_Toc159843098)

[Modello matematico 3](#_Toc159843099)

[1. Dati 3](#_Toc159843100)

[2. Variabili decisionali 3](#_Toc159843101)

[3. Vincoli 3](#_Toc159843102)

[4. Obbiettivo 4](#_Toc159843103)

[AMPL 5](#_Toc159843104)

[1. Dati 5](#_Toc159843105)

[2. Variabili decisionali 5](#_Toc159843106)

[3. Vincoli 5](#_Toc159843107)

[4. Obbiettivo 6](#_Toc159843108)

[Esempi 7](#_Toc159843109)

# Problema

**Ricollocazione nel bike-sharing**

Un’azienda che gestisce il bike-sharing ha il problema di ricollocare le bici ogni sera negli stalli dove si ha maggiore richiesta. In pratica ci sono *N* stalli, ogni stallo *i* ha un certo numero *ci* di bici parcheggiate in esso alla sera e ne vuole *si* parcheggiate nello stesso alla mattina. Il recupero delle bici avviene attraverso un furgone che ha una capacità massima di *K*. Sono poi noti i tempi di percorrenza *ti,j* tra le diverse coppie *i, j* ∈ *N* di stalli. Il problema da risolvere consiste nel cercare di pianificare le operazioni di ricollocazione delle bici in un tempo minimo.

Si formuli un modello matematico di questo problema, lo si traduca in AMPL e si risolva e commenti una particolare istanza del problema.

# Modello matematico

## Dati

* N = Numero di stalli.
* NODI = Insieme degli stalli.
* ARCHI = insieme degli archi (strade tra gli stalli).
* K = Capacità del furgone.
* ci con i ∈ NODI = Numero di bici presenti nell’i-esimo stallo la sera.
* si con i ∈ NODI = Numero di bici volute nello stallo i-esimo al mattino.
* ti,j con i, j ∈ NODI con i ≠ j = Tempo di percorrenza da i a j.
* pci ≔ ci - si con i ∈ NODI = Indica se nello stallo ci sono troppe bici (pci > 0) oppure se non ce ne sono abbastanza (pci < 0). Se è zero le bici sono già a posto, ed i sarà un nodo di transito (il circuito hamiltoniano deve passare da tutti i nodi).

## Variabili decisionali

* xi,j ∈ {0, 1} con i ≠ j = Indica se l’arco i, j fa parte del circuito hamiltoniano.
* yi ∈ {0, 1, …, N-1} = Indica la posizione del nodo i-esimo nella sequenza di visita dei nodi.
* fi ≥ 0, ≤ K = Indica la capacità del furgone nel nodo i-esimo.

## Vincoli

* Ogni nodo ha un solo arco in ingresso, ovvero ∑i ∈ NODI, i ≠ j xi,j = 1 Ɐ j ∈ NODI
* Ogni nodo ha un solo arco in uscita, ovvero ∑i ∈ NODI, i ≠ j xj,i = 1 Ɐ j ∈ NODI
* Non sono presenti sotto-circuiti, ovvero yj – yi ≥ xi,j + (1 – N)(1 - xi,j) Ɐ i ≠ j, j ≠ 1
* La y del primo nodo è fissata a 0, ovvero y1 = 0
* La f del primo nodo è fissata a pc del primo nodo se pc positivo altrimenti a 0, ovvero f1 = pc1 se pc1 > 0 altrimenti 0
* Se xi,j = 1 ⇒ fj – fi = pcj inevce se xi,j = 0 ⇒ fj – fi  ∈ [-K, K], ovvero fj – fi  ≥ pcjxi,j + (1 - xi,j)(-K) e fj – fi  ≤ pcjxi,j + (1 - xi,j)K

## Obbiettivo

* Il tempo totale deve essere minimo, ovvero min ∑i ≠ j ti,jxi,j

# AMPL

## Dati

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

## Variabili decisionali

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

## Vincoli

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

## Obbiettivo

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

# Esempi

ee