Dr. Luis A. Moncayo-Martínez
Optimisation of Manufacturing and Logistics Systems

- Research Topics
  - Simulation of manufacturing flexible systems.
  - Optimisation of inventory systems.
  - Mathematical modelling and computational implementation.
  - Parallel implementation of meta-heuristics.
- Research Projects
  - Optimizing the fiber optic deployment in rural communities.
  - Fast simulation of manufacturing system using R and C++.
  - Optimisation of service level of bike—sharing systems (PN2015-1234).
- Publications
  - https://orcid.org/0000-0003-4619-3808
  - https://scholar.google.com/citations?user=weu876YAAAAJ&hl=en
  - https://www.researchgate.net/profile/Luis-A-Moncayo-Martinez
- Contact:
  - luis.moncayo@itam.mx

$$\min TSCC = \xi \sum_{i=1}^{N} (\mu_i \sum_{j=1}^{N_i} c_{ij} y_{ij}) + h \sum_{i=1}^{N} (K_i z_i \sigma_i \sqrt{\zeta_i + \sum_{j=1}^{N_i} t_{ij} y_{ij} - \phi_i}) + h \sum_{i=1}^{N} (W_i \mu_i \sum_{j=1}^{N_i} t_{ij} y_{ij})$$

$$\min LT = \frac{1}{|D|} \sum_{i \in D} LT_i$$

Subject to:

$$LT_i = \sum_{j=1}^{N_i} t_{ij} y_{ij} + \max_{i':(i',i) \in E} \{LT_{i'}\} \quad \text{for} \quad i = 1,\ldots,N$$

$$\sum_{j=1}^{N_i} y_{ij} = 1 \quad \text{for} \quad i = 1,\ldots,N$$

$$\zeta_i + \sum_{j=1}^{N_i} t_{ij} y_{ij} - \phi_i \geq 0 \quad \text{for} \quad i = 1,\ldots,N$$

$$\phi_{i'} \leq \zeta_i \quad \text{for} \quad i = 1,\ldots,N\ i' : (i',i)$$

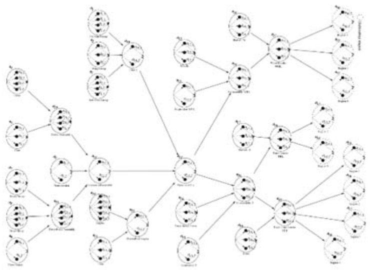
$$\phi_i \leq \Phi_i \quad \text{for all} \quad i \in D$$

$$K_i = \sum_{j=1}^{N_i} c_{ij} y_{ij} + \sum_{i':(i',i)} K_{i'}$$

$$W_i = \frac{1}{2} \sum_{j=1}^{N_i} c_{ij} y_{ij} + \sum_{i':(i',i)} K_{i'}$$

$$\phi_i, \zeta_i \geq 0 \quad \text{and integer for} \quad i = 1,\ldots,N$$

$$y_{ij} = \{0,1\} \quad \text{for} \quad i = 1,\ldots,N, \quad j \in i, \quad j = 1,\ldots,N_i$$



Algorithm 7 Calculate a solution set P

```
Require: G = {V, E}, R > 0, D > 0
Ensure: P = {s_1, ..., s_d, ...}
  Set r = 1
  Initialise S_ij, v_d, τ_ij
  if r ≤ R then
    Create the r^th river with D drops
    Set d = 1
    while d ≤ D do
      (first part of the algorithm, SC configuration)
      for i = 1 to i = I do
        Compute the probability p_ij ∀ j ∈ i, see (7.12)
        Select one option j based on p_ij, thus (7.4) is solved
        Solve constraints (7.5) and (7.6), thus T_i and C_i are set
        Update drop velocity v_d, see (7.13)
        Compute soil increments, ΔS_ij, see (7.14)
        Update soil of the selected option, j ∈ i, see (7.15)
      end for
      (second part of the algorithm, place safety stock)
      Compute safety stock cost to solve constraints (7.7)–(7.9)
      Calculate the in-transit inventory
      Calculate the LT_i for every stage i, see (7.3)
      Compute the solution s_d = (LT, IC), see (7.1)–(7.2)
      d++
    end while
    for all s_d do
      if s_d is non-dominated then
        P ← s_d
      end if
    end for
    Update soil on selected options j used to compute s_d ∈ P, see (7.16)
    r + +
  else
    P_R = P = {s_1, ..., s_d, ...}
  end if
```