DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

The Pickup and Delivery Problem with Scheduling at the Dock

Vitor A. A. Souza

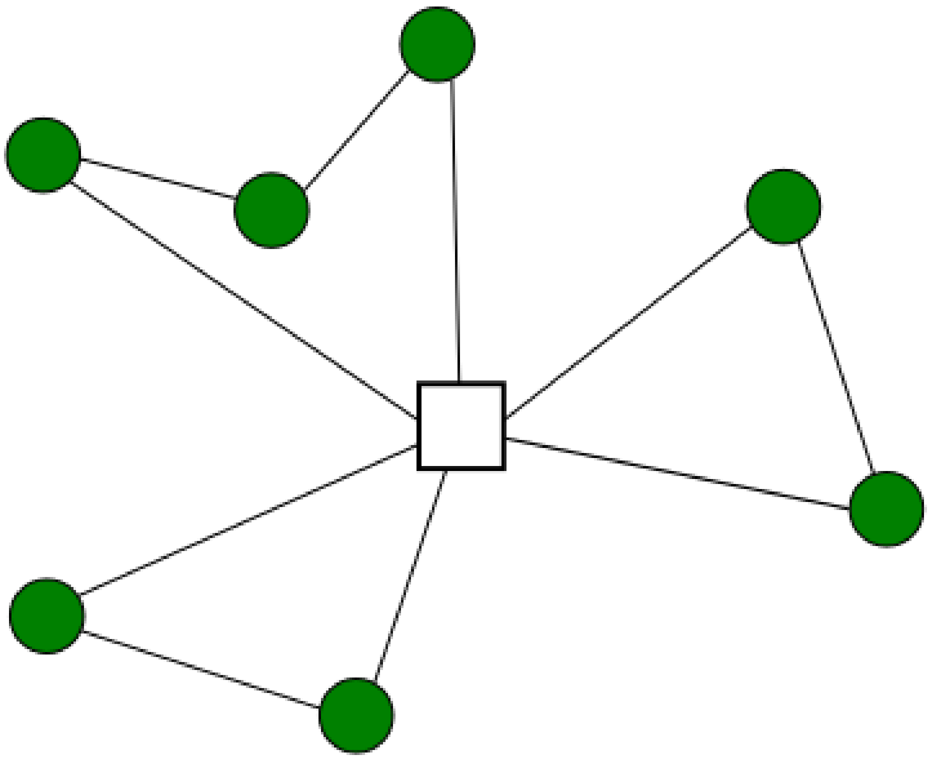
Geraldo Robson Mateus

Introduction

- It is an integration of the Pickup and Delivery Problem and the Cross-Docking Problem
- It is based on the Vehicle Routing Problem with Cross-Docking (VRPCD)
- As far as we know, this problem was not tackled in the literature yet

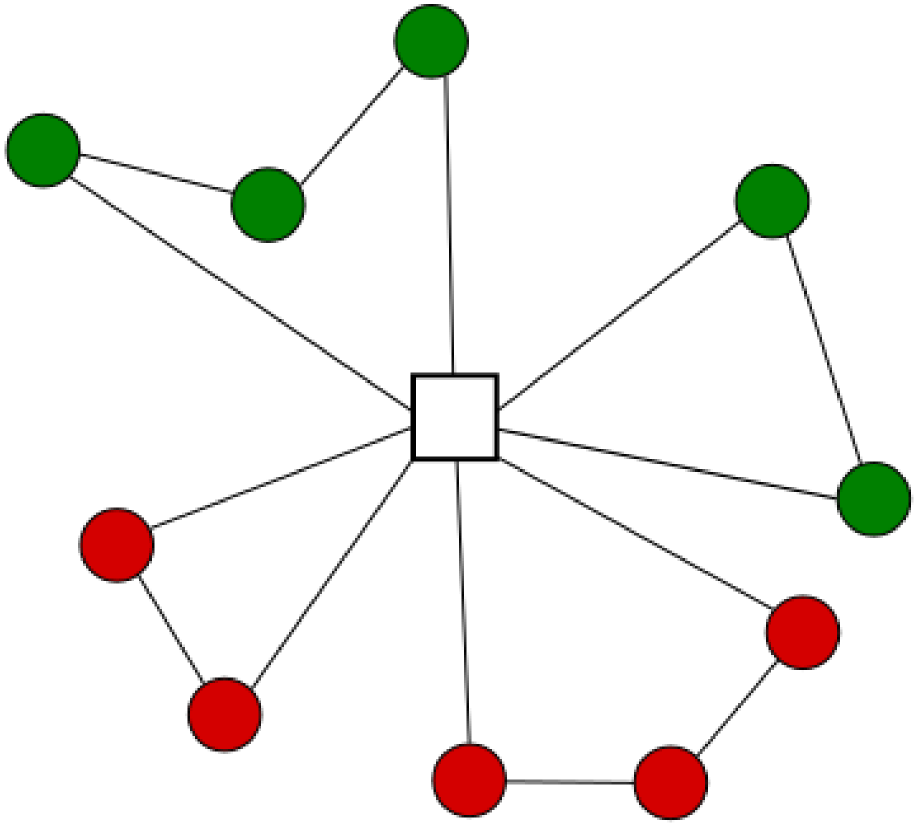
Introduction

- Classical Vehicle Routing Problem (VRP)



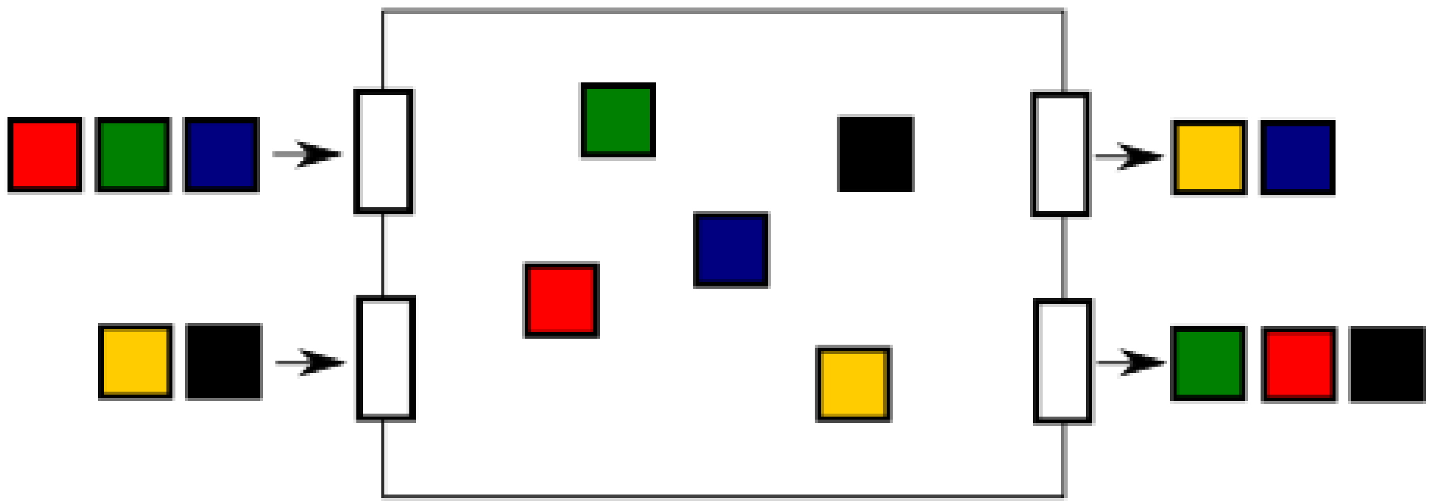
Introduction

- Pickup and Delivery Problem (PDP)



Introduction

- Cross-Docking Problem



Introduction

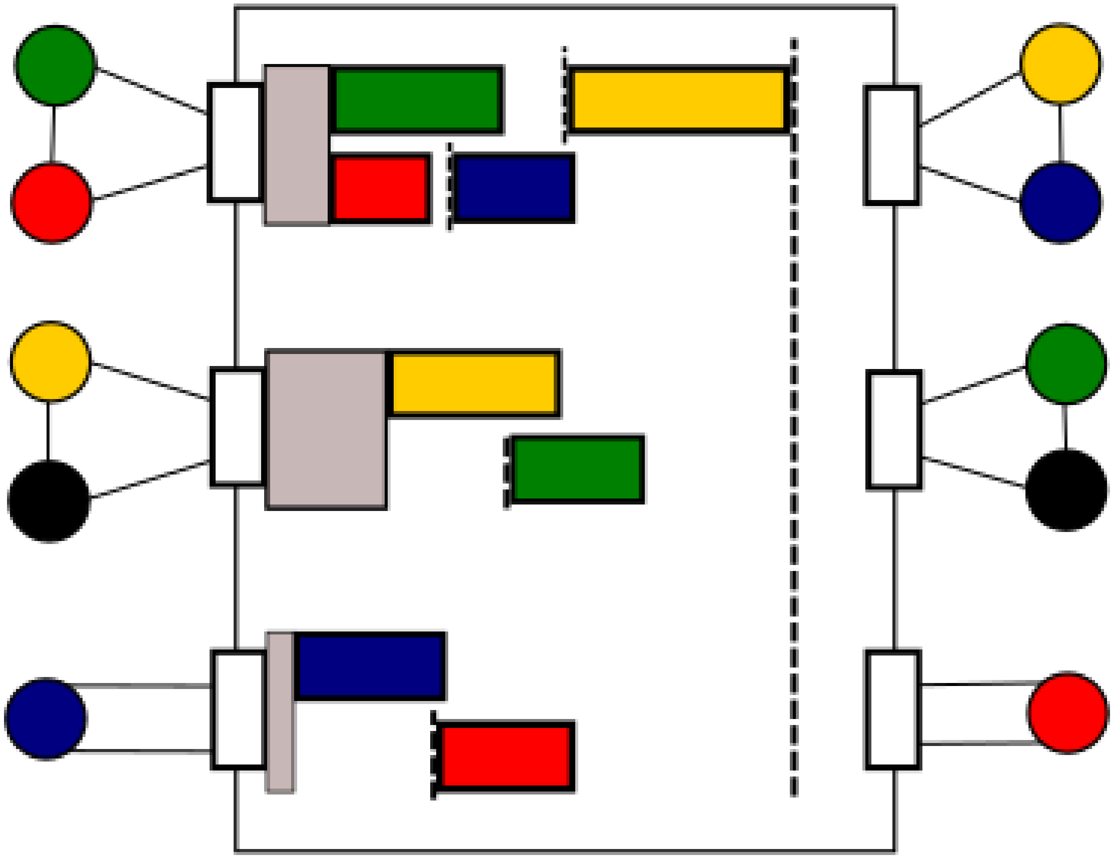
- Pickup and Delivery with Scheduling at the Dock
  - *n* requests one-to-one (supplier-customer pairs)
  - The set of vehicles that pickup the requests is the same that deliver them
  - We consider the time spent in the routing instead of distance traveled
  - The number of doors is the same of vehicles

Introduction

- Pickup and Delivery with Scheduling at the Dock
  - The objective considered in the scheduling is makespan
  - Each vehicle unloads their requests at the same time
  - The unloading and loading processes can be done at the same time in a vehicle, since it is at the dock

Introduction

- Pickup and Delivery with Scheduling at the Dock



Motivation

- Its practical application
- The VRPCD is still a very simplified version of the real problem

Column Generation

- The formulation is based on the set partitioning and it is indexed by the number of vehicles

- Each column generated represents a route

Column Generation

- The objective function

$$\min \sum_{k \in K} \sum_{p \in P_k^S} c_p \alpha_p^k + \sum_{k \in K} \sum_{p \in P_k^C} c_p \beta_p^k + t_{max}$$

Column Generation

- Routing related constraints

$$\sum_{p \in P_k^S} \alpha_p^k = 1, \qquad \forall k \in K$$

$$\sum_{p \in P_k^C} \beta_p^k = 1, \qquad \forall k \in K$$

$$\sum_{k \in K} \sum_{p \in P_k^S} a_{ip} \alpha_p^k = 1, \qquad \forall i \in S$$

$$\sum_{k \in K} \sum_{p \in P_k^C} b_{ip} \beta_p^k = 1, \qquad \forall i \in C$$

Column Generation

- Linking constraints

$$\sum_{p \in P_k^S} a_{ip} \alpha_p^k = y_i^k, \qquad \forall i \in S, \forall k \in K$$

$$\sum_{p \in P_k^C} b_{ip} \beta_p^k = z_i^k, \qquad \forall i \in C, \forall k \in K$$

Column Generation

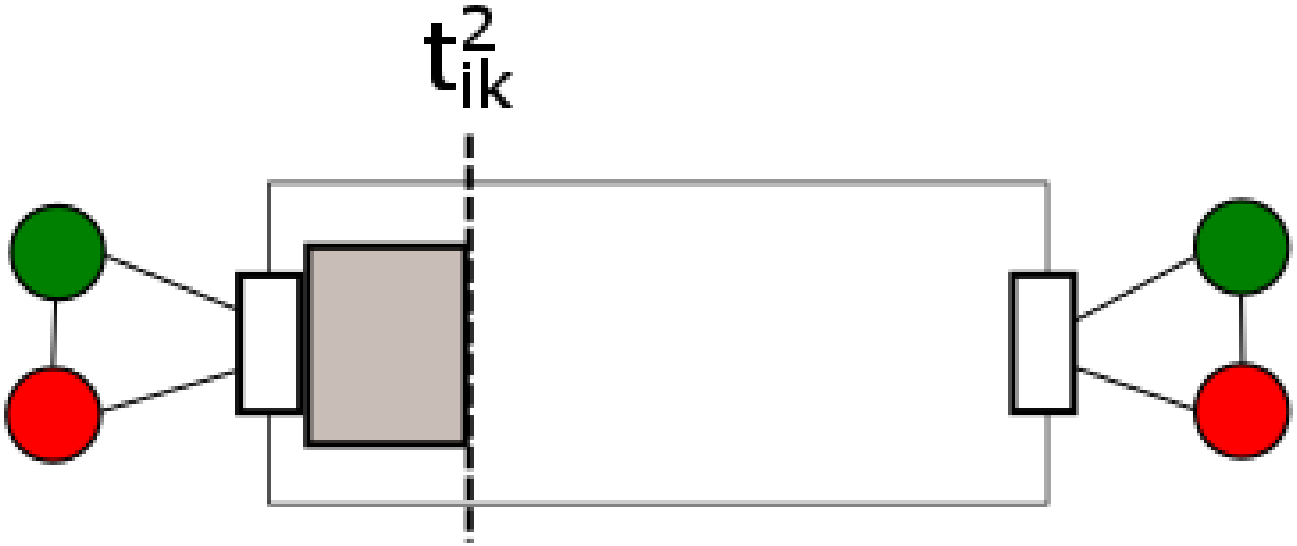
- Scheduling constraints

$$t_{ik}^{1} \geq t_{k}^{0} + \sum_{p \in P_{k}^{S}} c_p \alpha_p^k - M_1 \left(1 - y_i^k\right)$$

$$t_{k}^{1} \geq t_{ik}^{1}$$

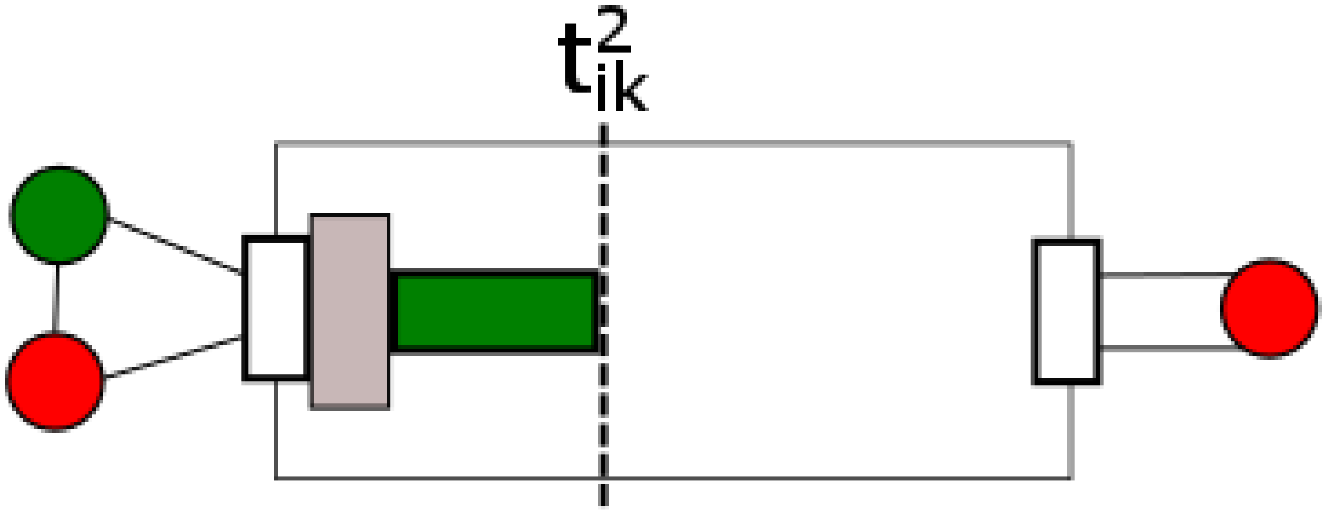
Column Generation

$$t_{ik}^2 \geq t_{ik}^1$$



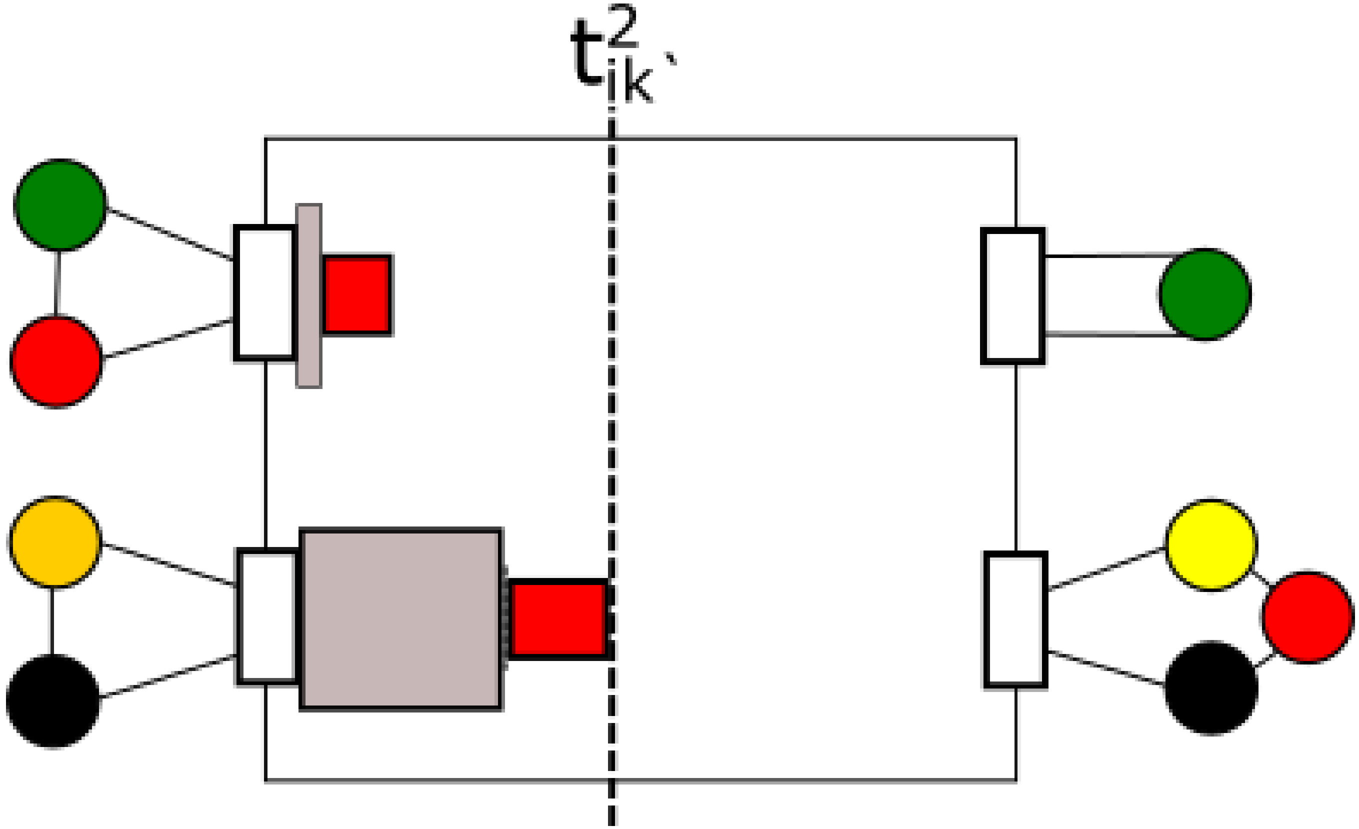
Column Generation

$$t_{ik}^{2} \geq t_{ik}^{1} + p_i - M_2 z_i^k - M_2\left(1 - y_i^k\right)$$



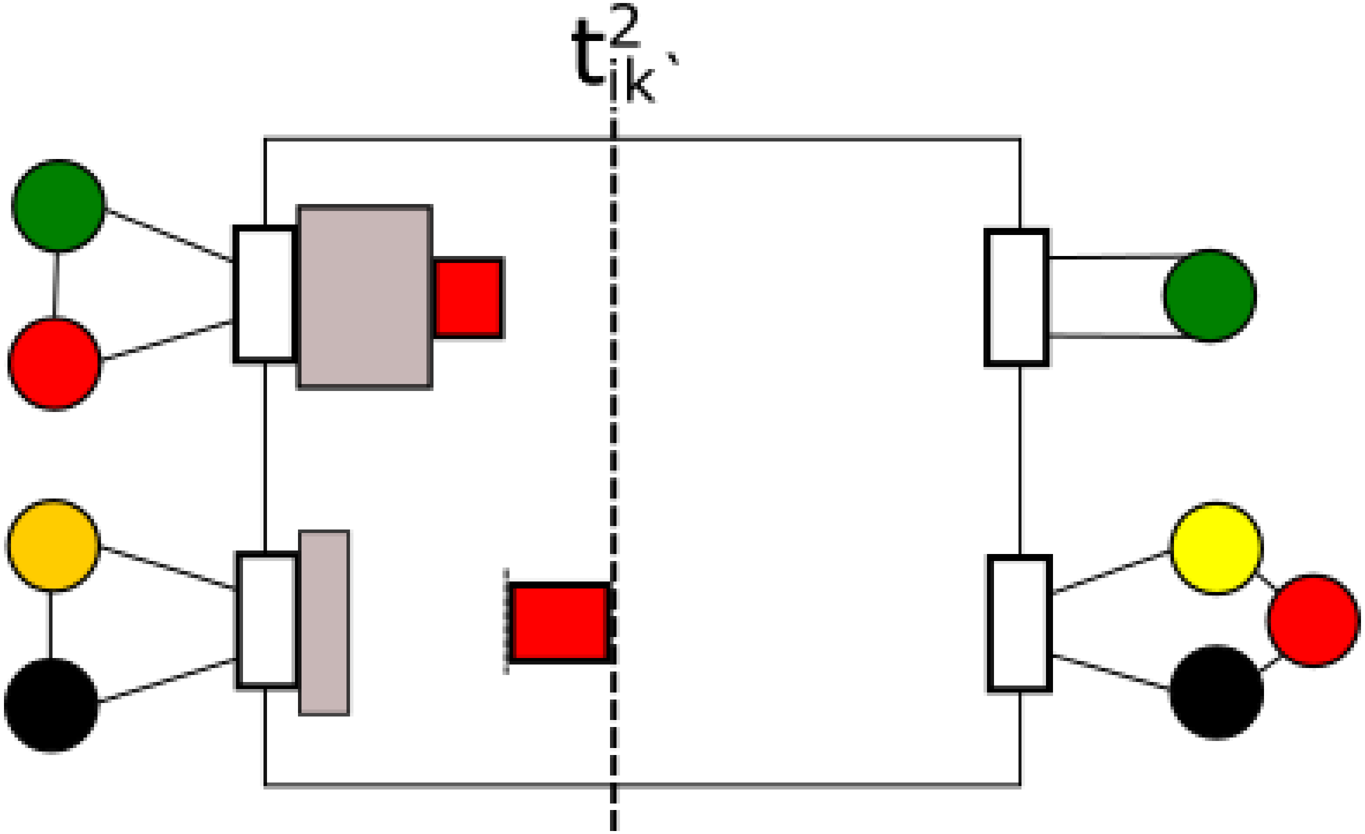
Column Generation

$$t_{ik'}^{2} \geq t_{k'}^{1} + p_{ikk'} - M_3\left(1 - z_i^{k'}\right) - M_3\left(1 - y_i^{k}\right)$$



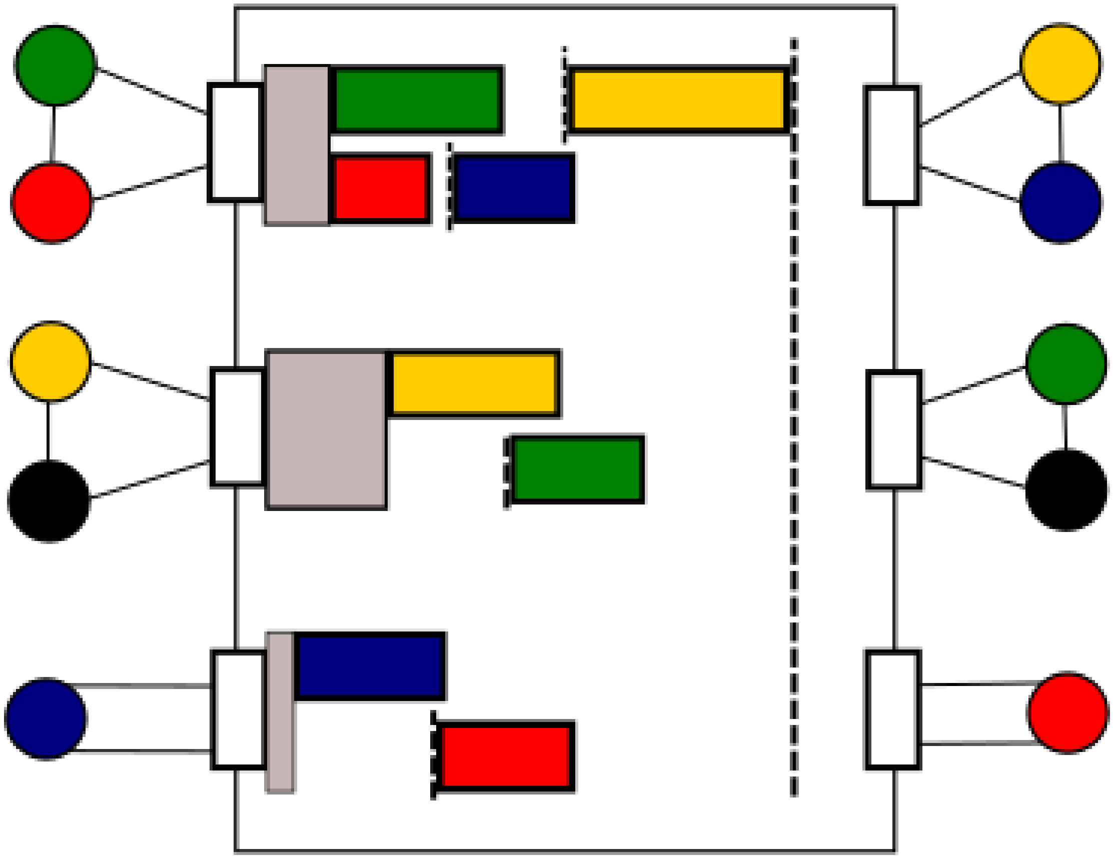
Column Generation

$$t_{ik'}^2 \geq t_{ik}^2 + p_{ikk'} - M_4\left(1 - z_i^{k'}\right) - M_4\left(1 - y_i^k\right)$$



Column Generation

$$t_{max} \geq t_{ik}^{2}$$



Column Generation

- Pricing subproblem
  - Elementary Shortest Path with Resource Constraints

- It is solved with dynamic programming (Feillet et al. [2004])

Computational Experiments

- The instances used in these experiments were based on the instances of Wen et al. [2009]

- Instances were generated with $|R| = \{5,7,10,12,15,18,20,22,25,27,30\}$ (5 instances of each size of requests)

Computational Experiments

- To check the quality of the results obtained, we have made a 2-Commodity Flow formulation for the problem

- To try to obtain primal solutions with the CG, we have converted the variables on the Master Problem to integer and solved with the Branch-and-Cut of CPLEX

Computational Experiments

	CG		2CF		
|R|	**dual bound**	**time(s)**	**int. relaxation**	**time(s)**	**distance (%)**
5	1005.231	0.04	887.176	0.01	13.31
7	1442.498	0.08	1340.108	0.02	7.64
10	1819.701	0.44	1674.040	0.06	8.70
12	2042.582	0.95	1821.609	0.11	12.13
15	2460.845	4.58	2241.795	0.26	9.77
18	2773.634	5.94	2475.960	0.40	12.02
20	2982.718	14.55	2699.756	0.51	10.48
22	3527.272	14.72	3214.250	1.02	9.74
25	3820.267	30.59	3494.215	1.55	9.33
27	4152.056	76.42	3748.552	2.30	10.76
30	4567.208	103.46	4202.981	3.76	8.67

Computational Experiments

	CG		2CF		
|R|	**primal bound**	**time(s)**	**solution**	**time(s)**	**distance (%)**
5	1192.070	0.07	1192.070	0.24	0.00
7	1629.125	0.24	1620.003	1.84	0.56
10	2035.261	1.52	2019.091	67.24	0.80
12	2344.108	8.19	2328.770	1111.38	0.66
15	2770.199	283.51	2746.495	3267.61	0.86
18	3068.111	620.86	3047.327	3601.81	0.68
20	3270.241	2144.92	3275.914	3602.80	-0.17
22	3838.860	2902.31	3860.496	3605.53	-0.56
25	4126.125	2993.45	4187.432	3605.86	-1.46
27	4474.185	3618.85	4635.888	3608.74	-3.49
30	4868.384	3372.78	5231.508	3609.63	-6.94

Computational Experiments

	CG	Best primals	
|R|	**dual bound**	**solution**	**dual/primal (%)**
5	1005.231	1192.070	84.33
7	1442.498	1620.003	89.04
10	1819.701	2019.091	90.12
12	2042.582	2328.770	87.71
15	2460.845	2746.495	89.60
18	2773.634	3047.327	91.02
20	2982.718	3270.241	91.21
22	3527.272	3838.860	91.88
25	3820.267	4126.125	92.59
27	4152.056	4474.185	92.80
30	4567.208	4868.384	93.81

Conclusion

- The Column Generation obtained the best dual bounds for all the instances in a reasonable amount of time

- It seems that the CG is a promising approach to solve the studied problem

Thank you!

vitor.andrade@dcc.ufmg.br