Meta-heuristics for Synchronized Multi-Zone Multi-Trip Pickup and Delivery Problems

Teodor Gabriel Crainic

TeodorGabriel.Crainic@CIRRELT.net

Phuong Khanh Nguyen, Michel Toulouse

ESG UQÀM
École des sciences de la gestion
Université du Québec à Montréal

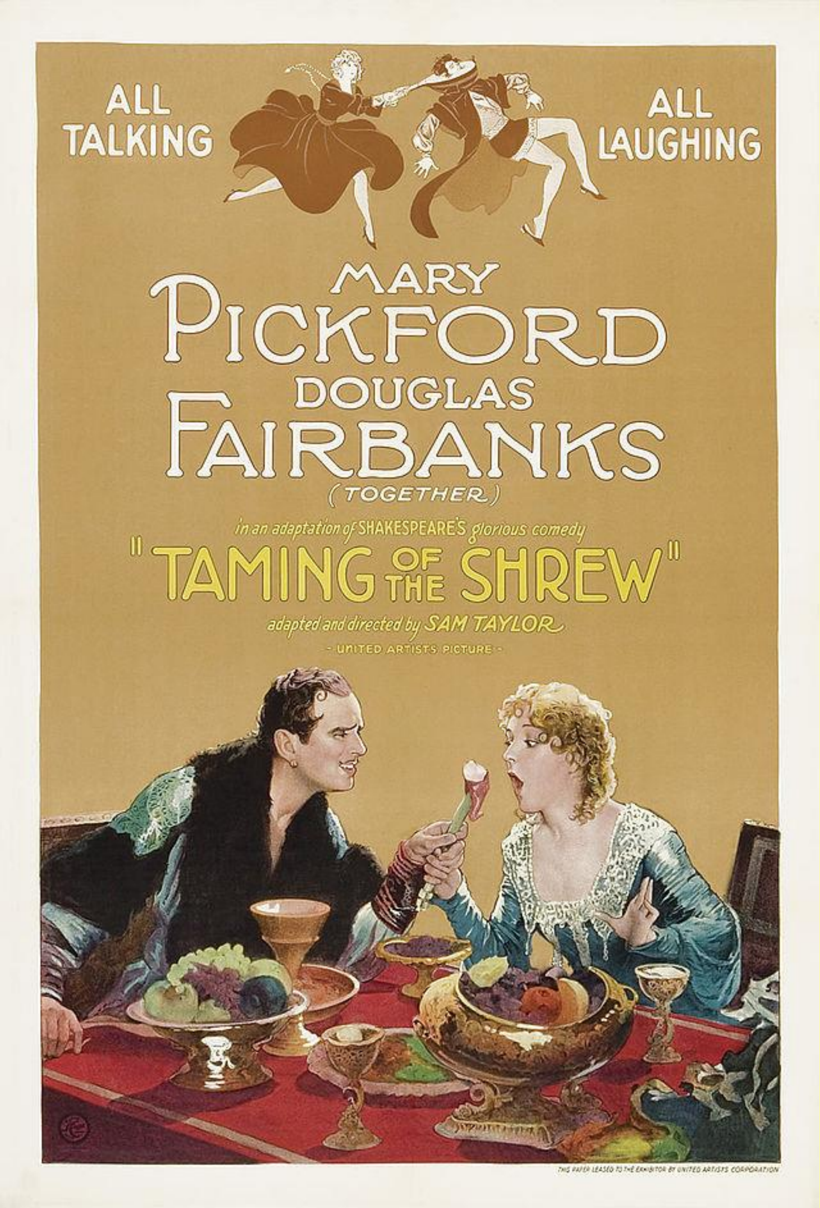
Chair in Logistics Management

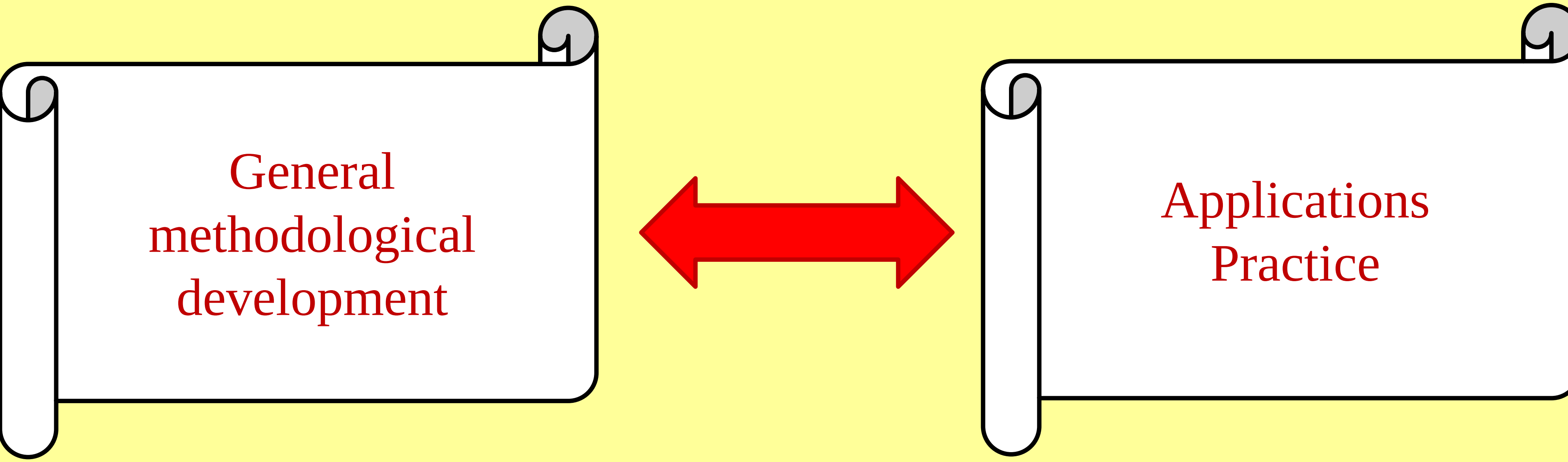
CIRRELT
Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

ROUTE 2014

The Taming of the Shrew
La mégère apprivoisée
La Bisbetica Domata
Imblânzirea scorpiei

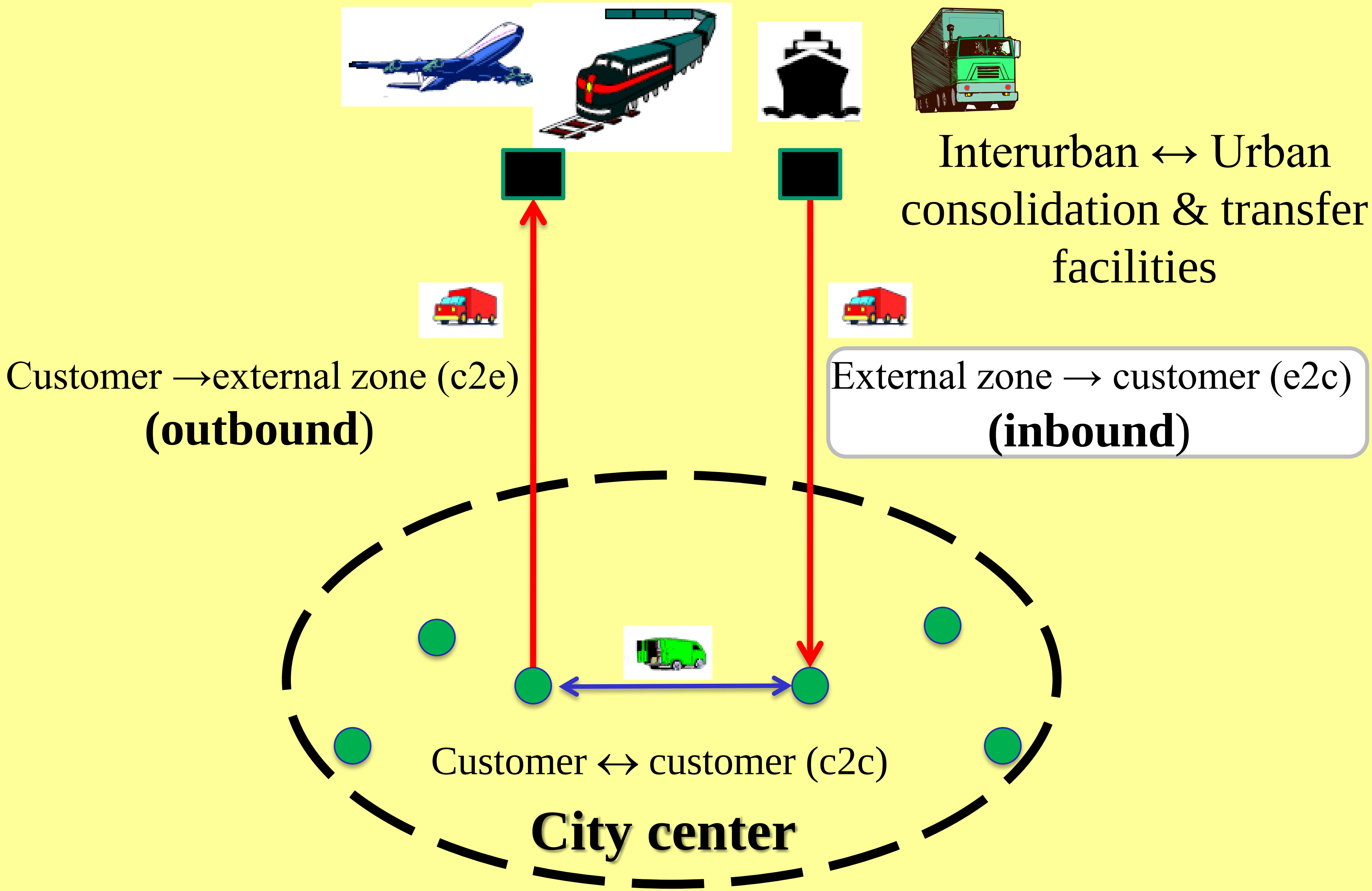

WORLD PREMIERE RECORDING OF THE ORIGINAL ORCHESTRAL SOUNDTRACK
LA BISBETICA DOMATA
THE TAMING OF THE SHREW
Music by
NINO ROTA


ALL TALKING
ALL LAUGHING
MARY PICKFORD
DOUGLAS FAIRBANKS
(TOGETHER)
in an adaptation of SHAKESPEARE'S glorious comedy
"TAMING OF THE SHREW"
adapted and directed by SAM TAYLOR
UNITED ARTISTS PICTURE


General methodological development
Applications Practice

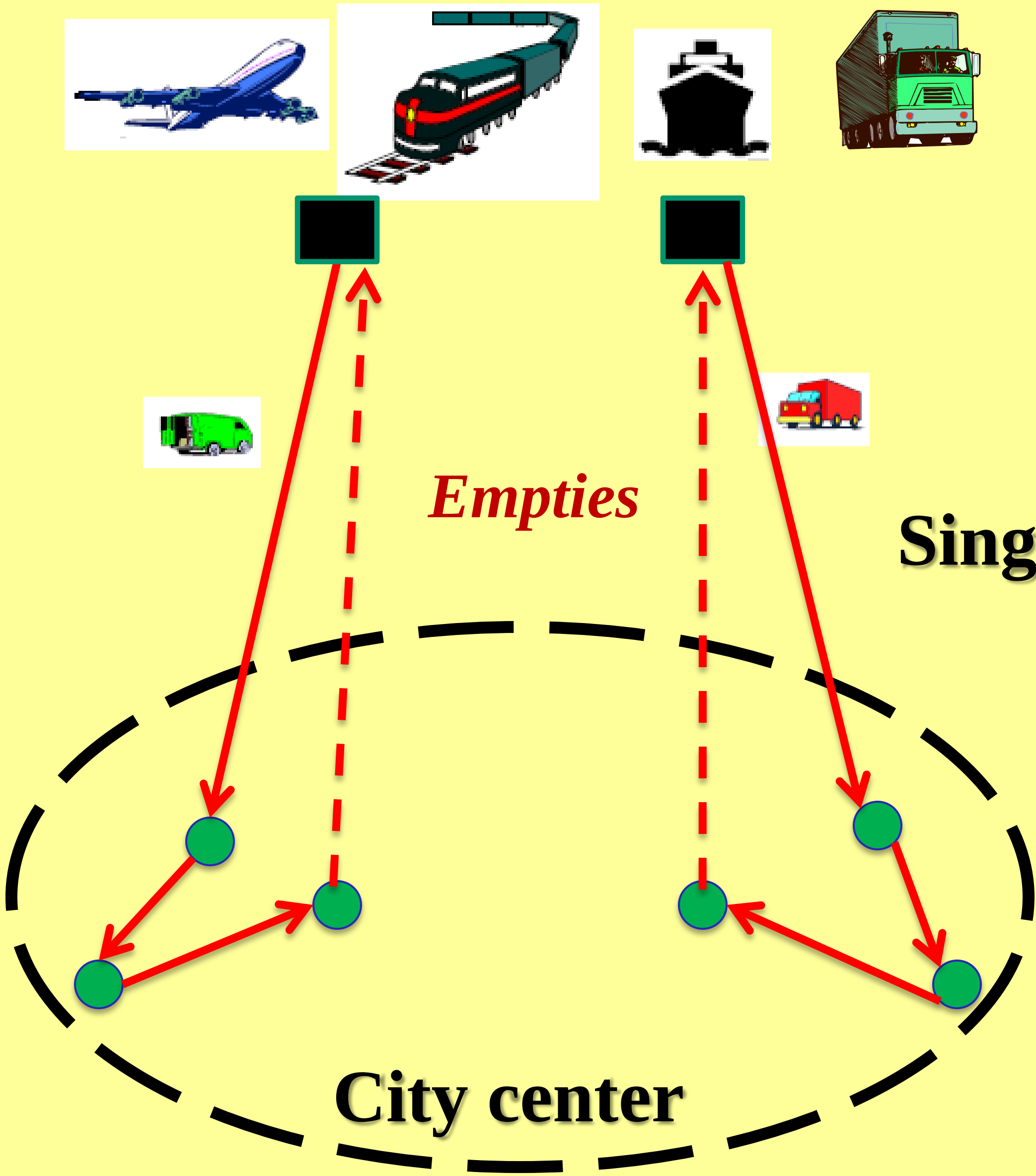
Outline

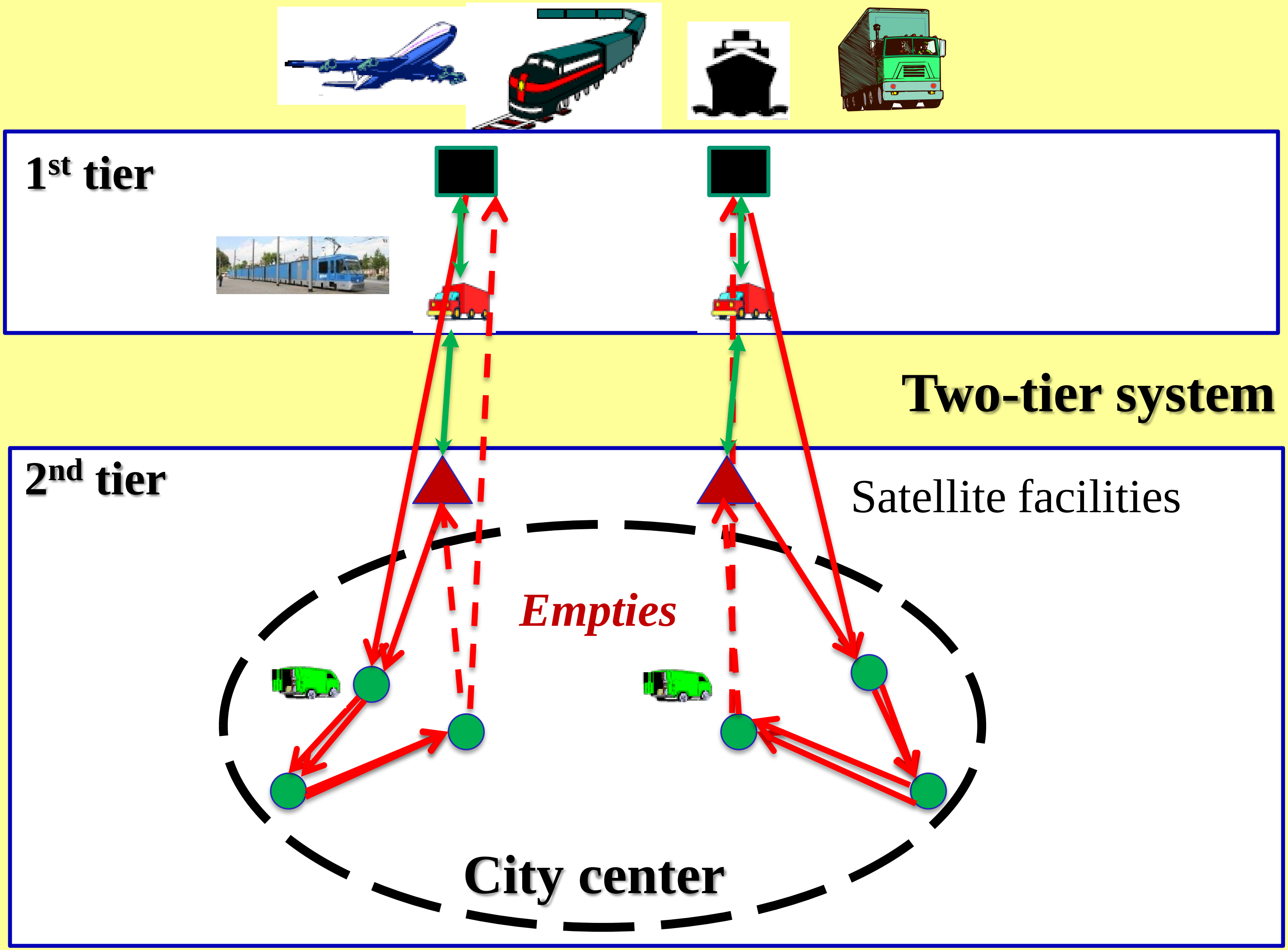
- Problem motivation and description
- The multi-zone, multi-tour pickup and delivery problem with time windows and synchronization
- Modelling
- A tabu search meta-heuristic
- Experimental results



Most City Logistics literature addresses ***inbound*** movements only

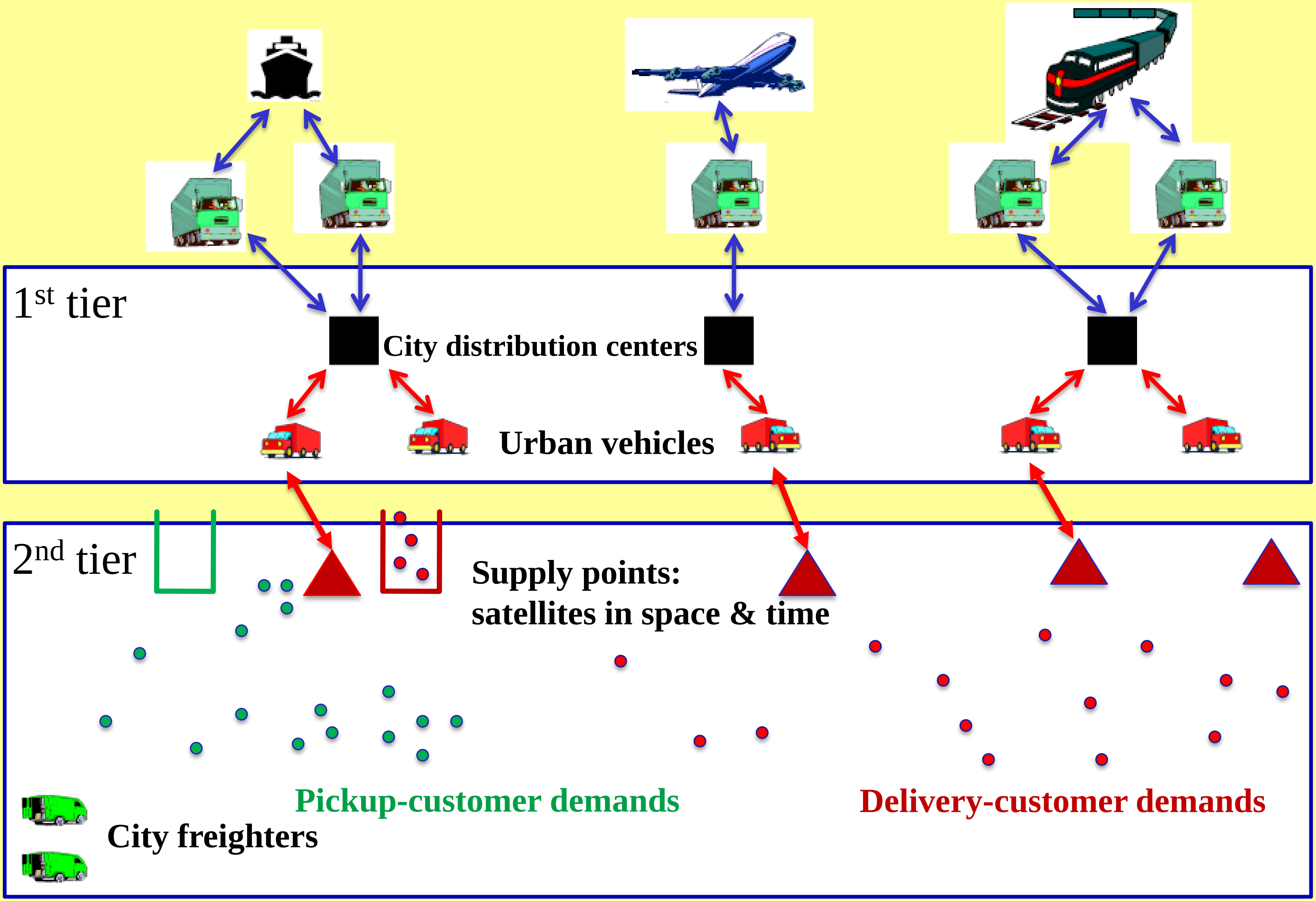
Single-tier system

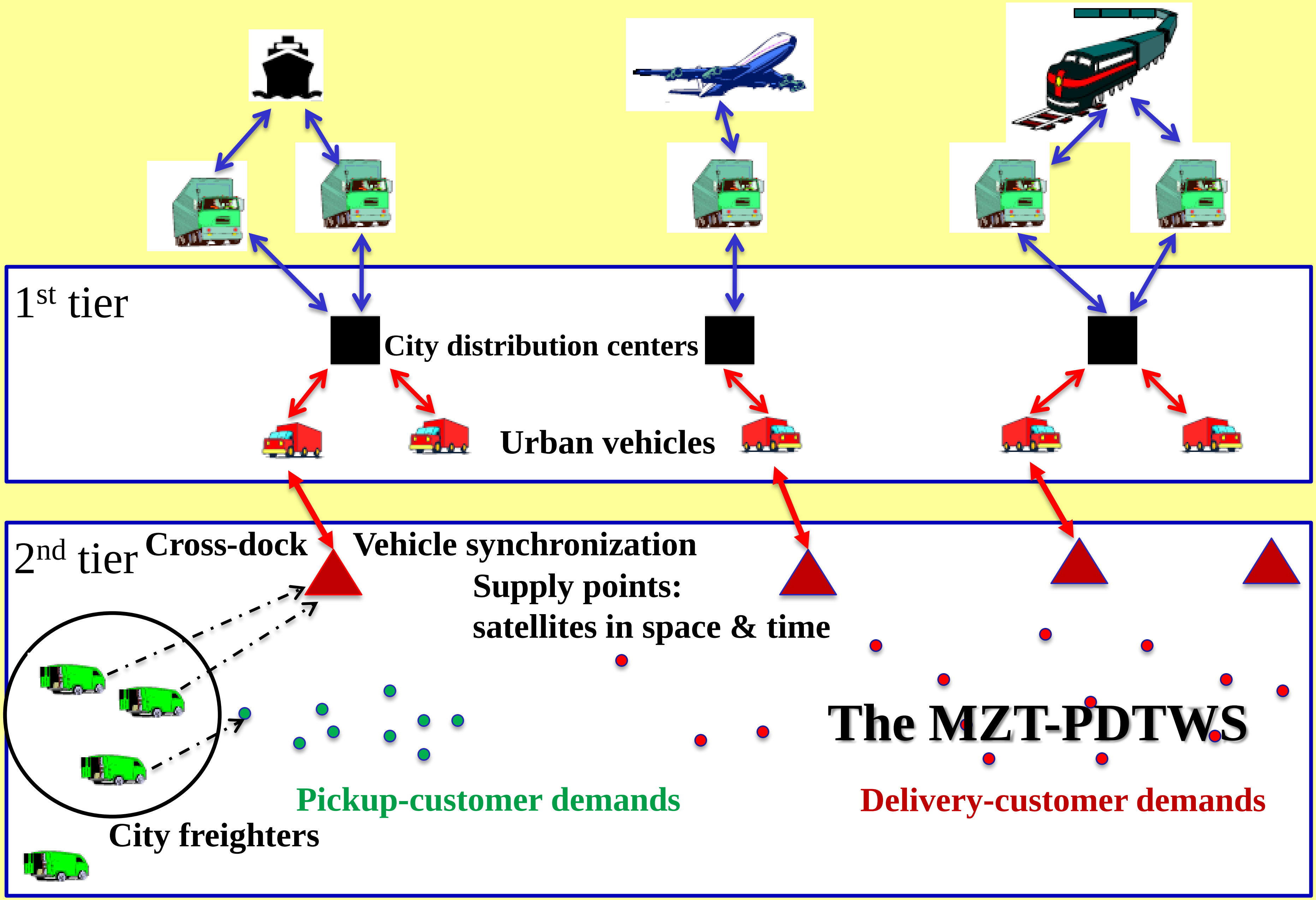



1st tier
Two-tier system
2nd tier
Satellite facilities
Empties
City center

Our Goals

- Focus on the second-tier routing problem
  - Demand time-dependency
  - Synchronization at satellites (hard time windows)
  - Customer time windows
  - Multiple tours, multiple zones
- Address more traffic types
  - e2c and c2e
- MZT-PDTWS, the multi-zone, multi-tour pickup and delivery problem with time windows and synchronization


1st tier
City distribution centers
Urban vehicles
2nd tier
Supply points:
satellites in space & time
Pickup-customer demands
Delivery-customer demands
City freighters


1st tier
City distribution centers
Urban vehicles
2nd tier
Cross-dock
Vehicle synchronization
Supply points:
satellites in space & time
The MZT-PDTWS
Pickup-customer demands
Delivery-customer demands
City freighters

Problem Definition

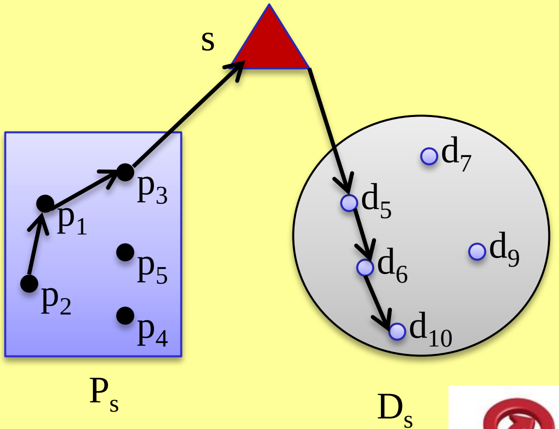
- A fleet of homogeneous vehicles
  - Capacity Q, fixed cost F
- Depot g
- Supply points $s \in S$, pickup-customer demands $p \in P$, delivery-customer demands $d \in D$
- Delivery-customer demand d
  - Demand q_d, service time $\delta(d)$, hard time window $[e_d, l_d]$
  - Serviced from **a given** supply point $s \in S$

Problem Definition (2)

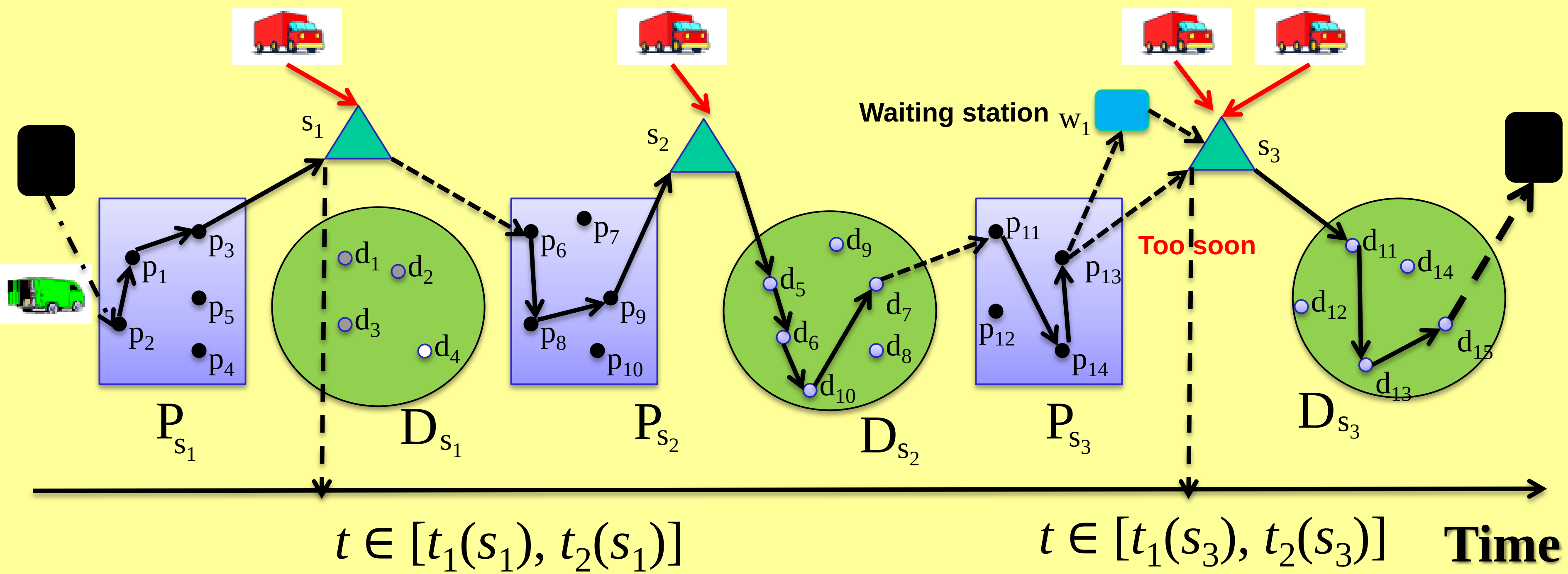
- Pickup-customer demand *p*
  - Demand q_p, service time $\delta(p)$, hard time window $[e_p, l_p]$
  - Serviced from a supply point **to be selected** in $S_p \subseteq S$
- Each supply point *s* services
  - Set of **given** delivery-customer demands D_s
  - Set of pickup-customer demands P_s **to be determined**
  - Unloading time $\delta_1(s)$, loading time $\delta_2(s)$
  - **No wait, hard time window** $[t_1(s), t_2(s)]$ for both unloading and loading

Problem Definition (3)

- Operation Strategy?
  - Many possibilities to interlace e2c and c2e activities
  - Each requiring different operations at satellites
  - More flexibility = More efficiency (less vehicles), but
  - Aim for "simple" satellite operations & management
- **Pseudo-Backhaul** strategy
  - A delivery or pickup phase must be completed before another can start
- **LIFO** loading & unloading

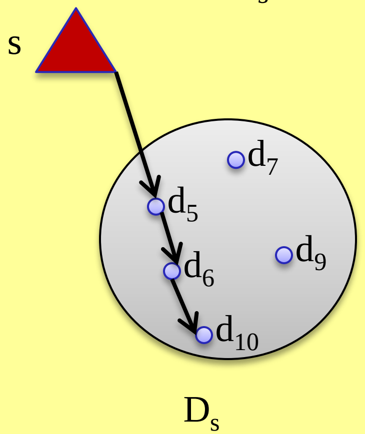


Waiting Stations $w \in W$

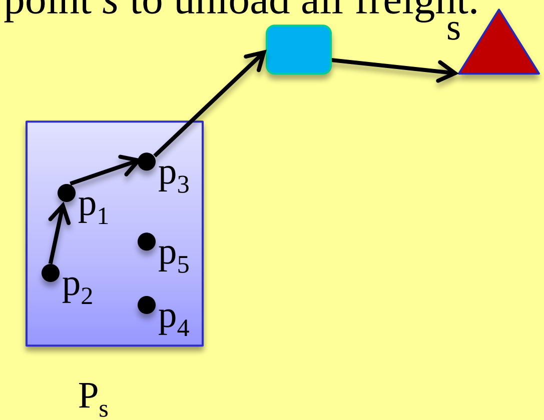


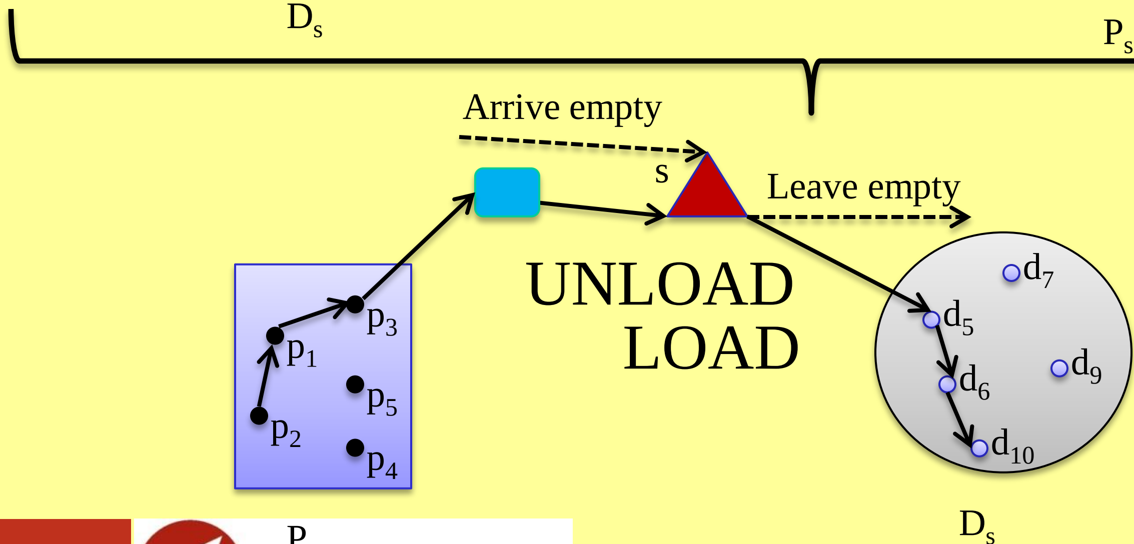
Route Building Blocks

- Delivery leg: partial route run by a vehicle that starts at a supply point *s*, loads freight, delivers to one or several delivery-customer demands in D_s.



- Pickup leg: partial route run by a vehicle that visits one or several pickup-customer demands *p* in P_s to load freight, may wait at a waiting station, and ends at the supply point *s* to unload all freight.





(Pickup leg, *s*)

(*s*, Delivery leg)

(Pickup leg, *s*, Delivery leg)

The MZT-PDTWS

- Minimize the sum of
  - Fixed vehicle cost
  - Routing cost
- Assign pickup demand customers to one of permitted supply points
- Satisfy time-dependent demand with its time windows
- Visit supply points within time windows
- Vehicle capacities

A Tabu Search Meta-heuristic

- Two decision levels:
  - High: vehicles (routes) are assigned to supply points
  - Low: pickup/delivery legs created by assigning pickup/delivery-customer demands to vehicles
- Both decisions are adjusted along the search by using leg and routing neighborhoods
  - Usage dynamically adjusted
- ***Control*** procedure: control dynamically the selection of neighborhood types
- ***Diversification*** strategy guided by an elite set and a frequency-based memory

Generate an initial feasible solution z

Elite set $E \leftarrow \emptyset$; $z_{\text{best}} \leftarrow z$

Probability of selecting routing neighborhood with respect to leg neighborhood $r \leftarrow 1$

STOP $\leftarrow 0$

Repeat

{ ***Select*** a neighborhood based on r

Explore the selected neighborhood of z & ***identify*** the best solution z' in $N(z)$

if z' is better than z_{best} **then** { $z_{\text{best}} \leftarrow z'$, Add z_{best} to the elite set E; *Manage E* }

$z \leftarrow z'$

if z_{best} not improved for IT_{CNS} iterations **then**

{ **if** z_{best} not improved after C_{CNS} consecutive executions of ***Control*** procedure **then**

{ **if** $E \leftarrow \emptyset$ **then** STOP $\leftarrow 1$

else { Select randomly z (and remove it) from E

Diversify the current solution z }

}

else { Call ***Control*** procedure to update the value of r

$z \leftarrow z_{best}$ }

}

} **Until** *STOP*

$z_{best} \leftarrow Post\_Optimization(z_{best})$

Search Space

- Feasible and infeasible solutions
  - Violations of vehicle capacity $q(p)$, supply point time window $w_s(p)$, customer demands time window $w_c(p)$
- Weighted fitness function

$$f(p) = c(p) + \alpha_1 \, q(p) + \alpha_2 \, w_c(p) + \alpha_3 \, w_s(p)$$

- Penalty parameters α_1, α_2, α_3 dynamically adjusted with respect to the evolution of violations
  - (Cordeau el al. 2001)

Initial Solution

- Assign each pickup customer demand to one of the supply points in its set
  - "Balance" at each supply point, the total incoming load (picked up at p customers) and K_s the total load that must be moved out of the supply point and delivered to d customers
- Build routes with those assignments

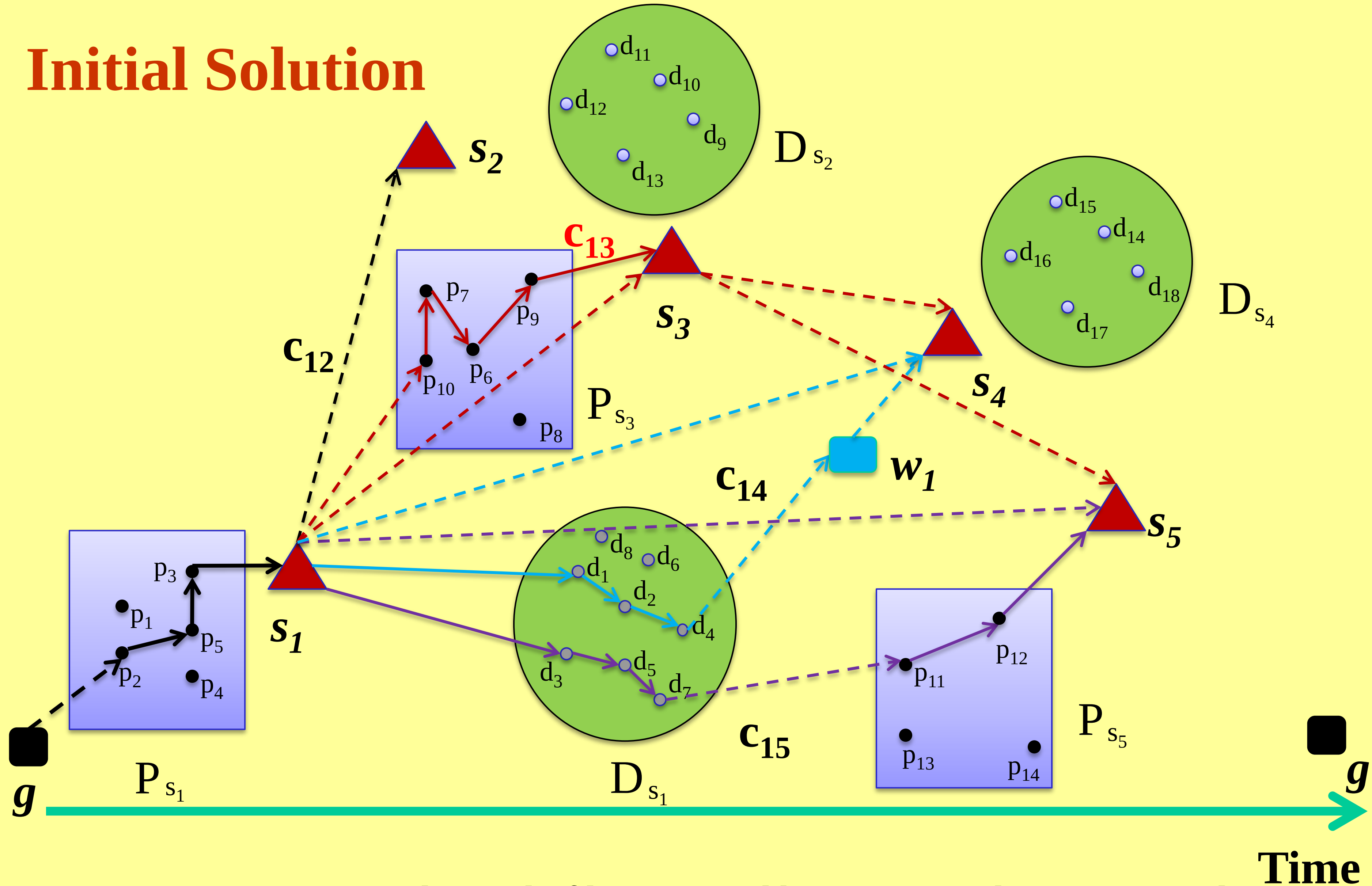
Initial Pickup Customer Assignment

- Pickup-customer demands handled in random order
- Each pickup-customer demand p assigned to a supply point $s \in S_p$ such that the value of K_s is respected
  - s is closest to p if this assignment does not violate K_s
  - s is selected randomly from $S_p - \{closest\ to\ p\}$, otherwise

Initial Vehicle Routes

- Build each vehicle route sequentially in two phases:
  - Determine the first supply point for the current vehicle: unrouted customer demands + the earliest opening time
  - Create legs sequentially by applying a greedy algorithm

Initial Solution



c_{ij}: Average cost per unit demand of legs created between supply points s_i and s_j

Two Types of Neighborhoods

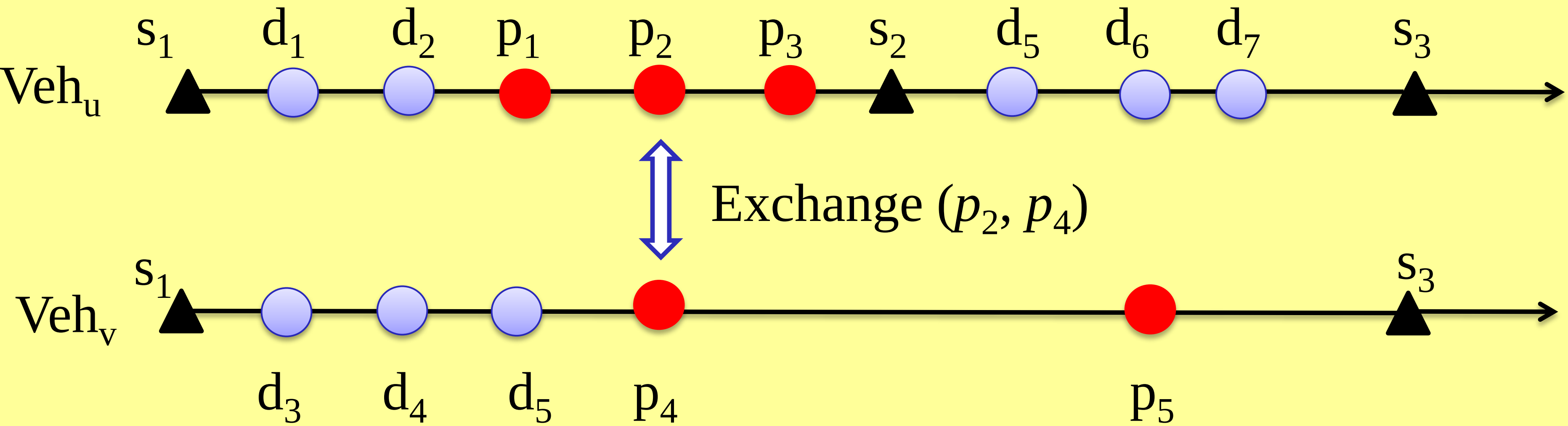
- Routing neighborhoods
  - Improve routing by using different intra and inter route neighborhoods commonly used in the VRPTW literature
- Leg neighborhoods
  - Move supply points (and associated legs) between vehicle routes

Routing Neighborhoods

- Work on the sets of pickup and delivery legs separately
- Three types, all involving two customer demands
- Relocation move
  - One of two customer demands is taken from its current position and inserted after the other one
- Exchange move
  - Two customer demands are swapped
- 2-opt move
  - Customer demands belong to the same leg: 2-opt
  - Customer demands belong to different legs: 2-opt*

Routing Neighborhoods & Pickup Customers

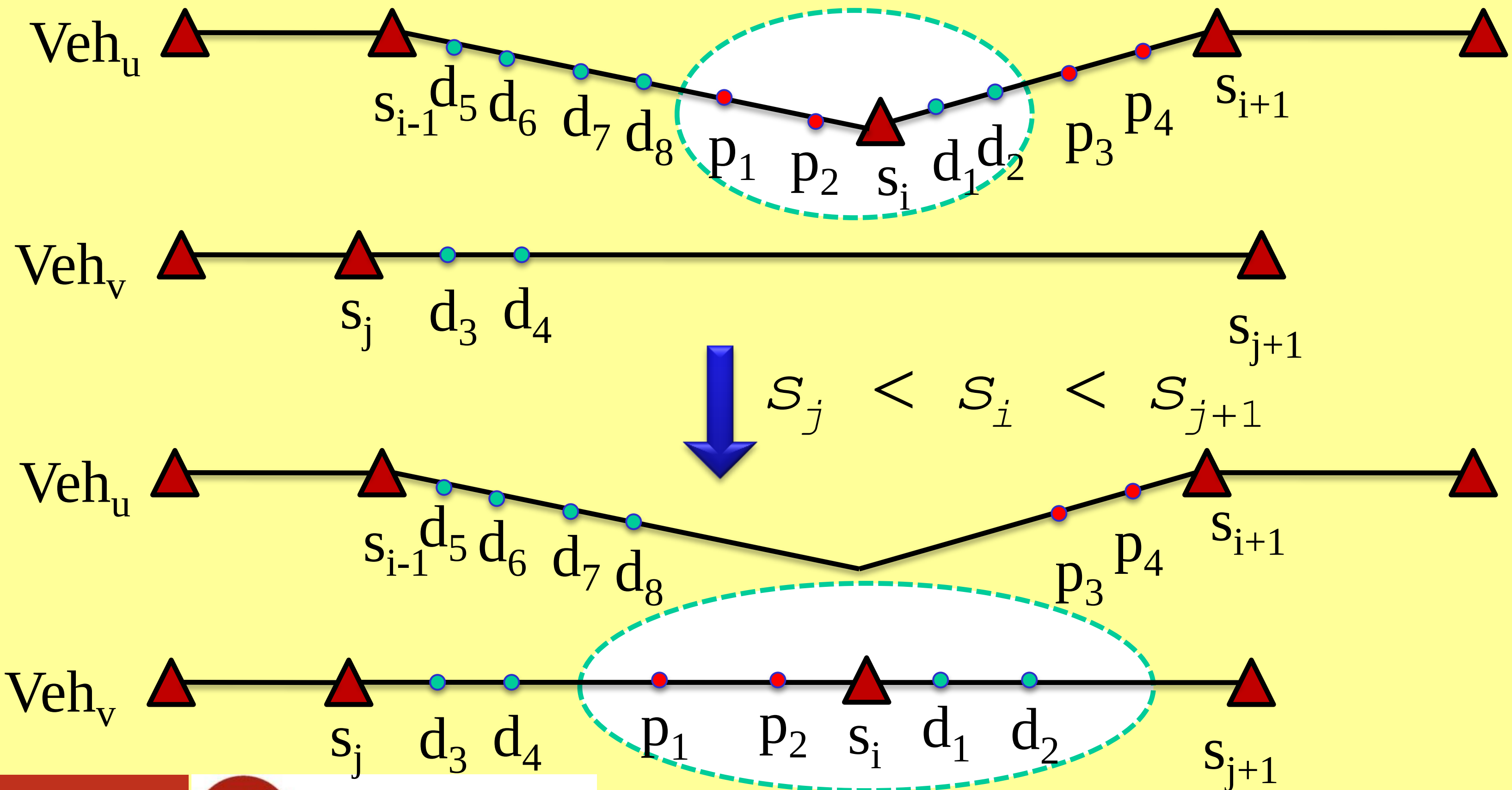
- Pickup-customer *p* reassigned to the next supply point *s* in the new route, if $s \in S_p$ (penalty adjustments, if needed)



Pickup-customer demand *p*	S_p	Current assignment
p_1	s_1, s_2	s_2
p_2	s_2, s_3	s_2
p_3	s_2	s_2
p_4	s_1, s_2, s_3	s_3
p_5	s_1, s_2, s_3	s_3

Leg Neighborhoods

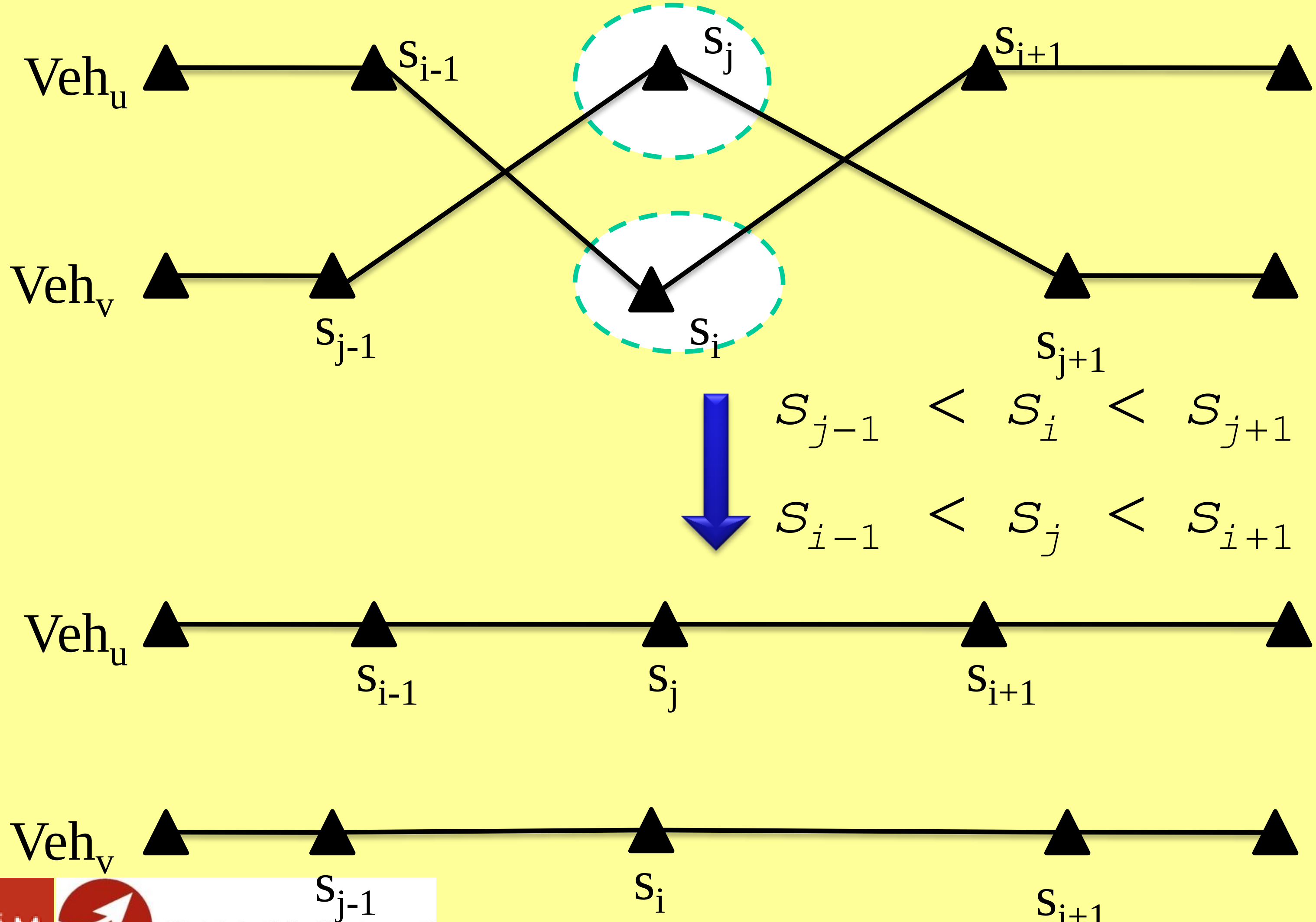
- Relocate supply point: remove a supply point **and** its legs (customer demands it services) from a route and insert them into another route



CIRRELT

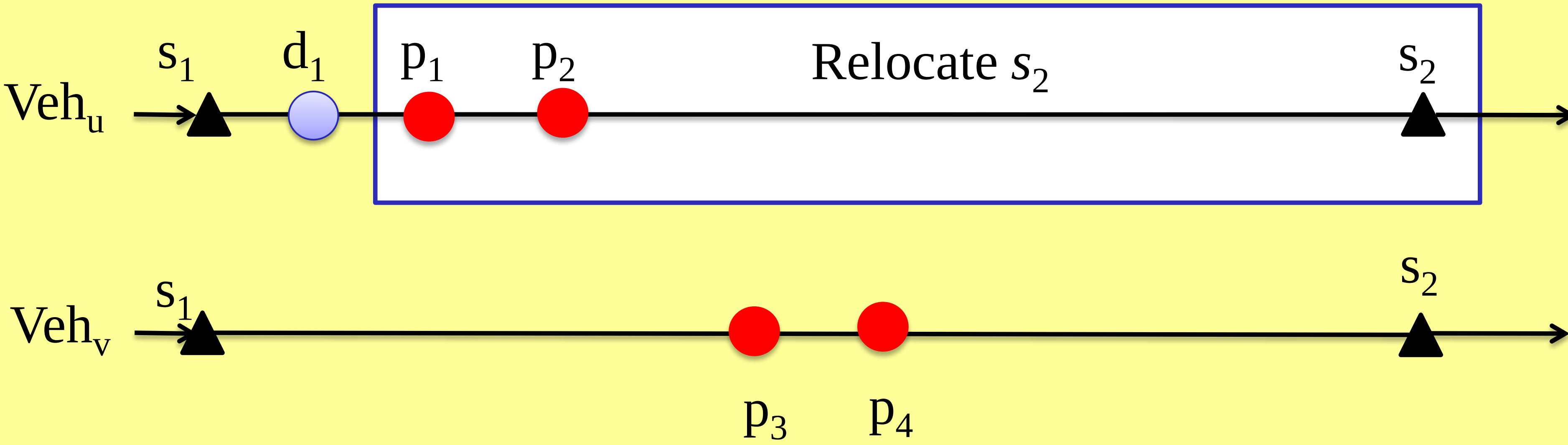
Leg Neighborhoods (2)

- Exchange supply points and legs between routes



Leg Neighborhoods (3)

- Reassignment of pickup-customer demands to supply points
- Concatenation of two pickup/delivery legs when assigned to the same supply point



Handling Two Types of Neighborhoods

- One neighborhood is selected at each iteration
- All neighborhoods start with the same probability of being selected
- The probability of selecting supply point neighborhoods decreases in time (the ***Control*** procedure)

Diversification

- Capitalize on the best attributes obtained so far
- Provide a certain level of diversity of the search.
- Elite set: best (& diversified) solutions identified so far
- Frequency memory: used arcs & supply point assignment to pickup customer demand
- Procedure
  - Take a solution from the elite set
  - Perturb this solution by removing arcs frequently used and introducing little seen assignments
  - Proceed by penalizing move evaluations

Post optimization

- Supply-point improvement of pickup-customer demands to supply point assignments
  - Pickup-customer demands are handled in random order
  - Assign each pickup-customer demand *p* to its unassigned supply point $s' \in S_p$, then re-route *p* (by the cheapest insertion); Keep the best one
- Leg improvement of routing
  - Intra-route: 2-opt of Lin (1965) and Or-opt of Or (1976)
  - Inter-route: λ-interchange of Osman (1993) [λ =1,2] and CROSS-exchange of Taillard et al. (1997)

Experimental Results

- 90 instances

Problem set	BH	#Customers	#Supply points	#Waiting stations	#Supply points available for each pickup customer
A1	{0.1, 0.3, 0.5}	400-800	4	4	1-2
A2		400-800	8	4	1-2
B1		1600-3200	16	16	1-3
B2		1600-3200	32	16	1-3
C1		3600-7200	36	36	1-4
C2		3600-7200	72	36	1-4

- Runs on a 2.8 GHz Intel Xeon with 16 GB of RAM

Design Alternatives

- Calibration of parameters
  - Generally defined as functions of problem size
- Diversification, elite set, memory are important

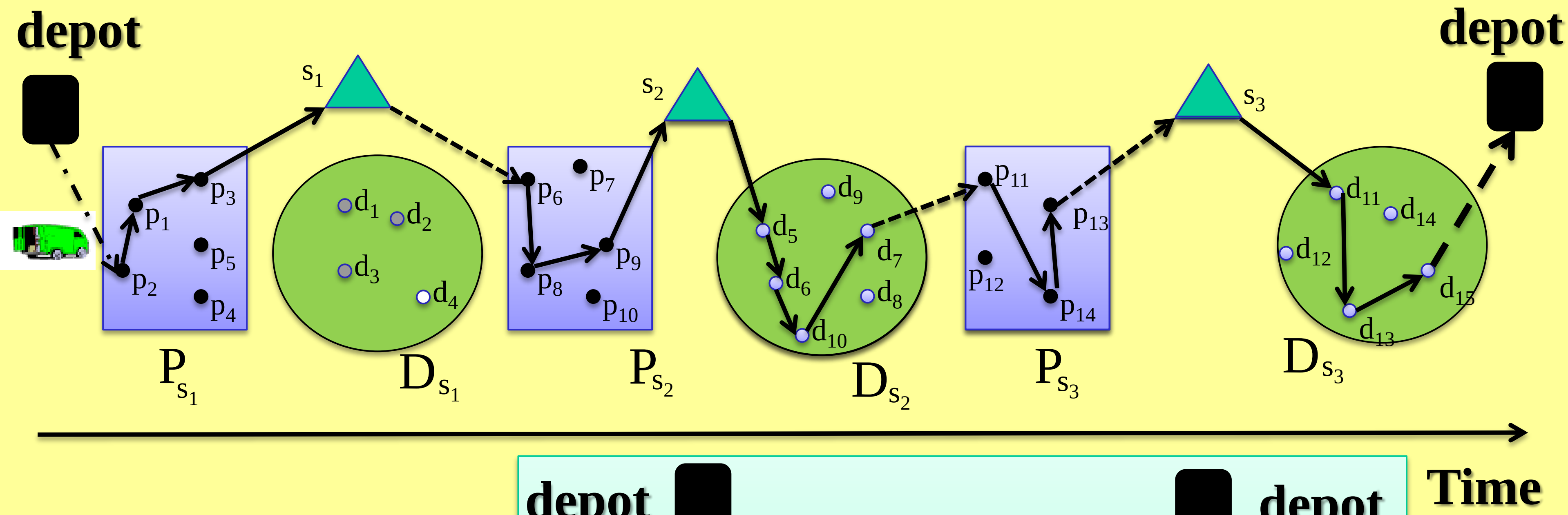
90 instances		
Without Diversification	**With Diversification**	
	Elite set	Elite set & frequency-based diversification strategy
104064.61	103252.92	102524.49
	-0.78%	-1.48%

Numerical Results

Set	Best 10	Avg 10	#Vehicles	DM(%)	PD(%)	Time (min)
A_1	21286.18	21445.59	22	29.73	56.64	37
A_2	18677.89	18832.46	17	30.60	55.44	21
B_1	80395.99	80574.20	50	29.65	47.38	145
B_2	75167.19	75317.25	41	25.11	47.23	112
C_1	214930.60	215146.80	103	23.41	47.48	395
C_2	204689.10	204982.00	93	23.10	45.11	224
Avg	**102524.49**	**102716.40**	**54**	**26.94**	**49.88**	**156**

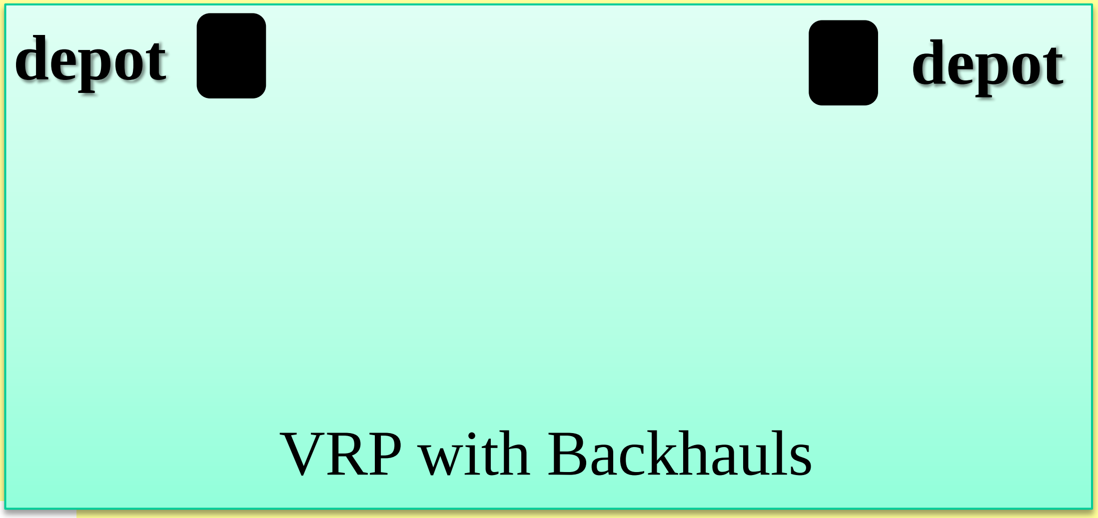
- DM (%): time % vehicles move directly to supply points without waiting stations
- PD (%): time % vehicles both unload and load once they arrive at supply points

Compared with the VRP with Backhauls



Minimize:

1) Number of vehicles
2) Routing cost



ESG UQÀM Chair in Logistics Management CIRRELT

VRP with Backhauls and Time Windows

- Gelinas et al. (1995): 15 instances (100 customers)

Authors	Method	CNV	CTD
Thangiah et al. (1995)	2-phase heuristic	274	24051.9
Potvin et al. (1996)	Genetic	267	23317.1
Reimann et al. (2002)	Ant system	265	23514.93
Reimann and Ulrich (2006)	Ant colony optimization	261	23942.44
Ropke and Pisinger (2006)	LNS	259	23416.81
Our work (F=0)	**Tabu**	**263**	**23395.51**

- Competitive with respect to total distance, outperforming four out of the five meta-heuristics (average gap =1.08%, maximal gap = 2.81% and a minimal gap = -0.34%)

VRP with Backhauls and Time Windows (2)

Authors	Method	CNV	CTD	
Thangiah et al. (1995)	2-phase heuristic	274	24051.9	
Potvin et al. (1996)	Genetic	267	23317.1	
Reimann et al. (2002)	Ant system	265	23514.93	
Reimann and Ulrich (2006)	Ant colony optimization	261	23942.44	
Ropke and Pisinger (2006)	LNS	259	23416.81	
Our work (F=0)	**Tabu**	**263**	**23395.51**	
F = avgcost		261	24084.11	2.94%
F = 1.1 * avgcost		261	24204.43	3.45%
F = 1.2 * avgcost		261	24232.07	3.57%
F = 1.3 * avgcost		261	24152.52	3.23%

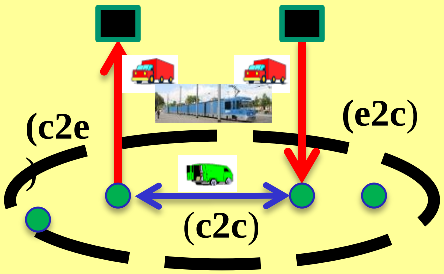
VRP with Backhauls (Without Time Windows)

- Compared with published tabu search methods
- Two instance sets:
  - Goetschalckx & Jacobs-Blecha (1989): 62 instances ([25, 150] customers)
  - Toth and Vigo (1997): 33 instances ([21, 100] customers).

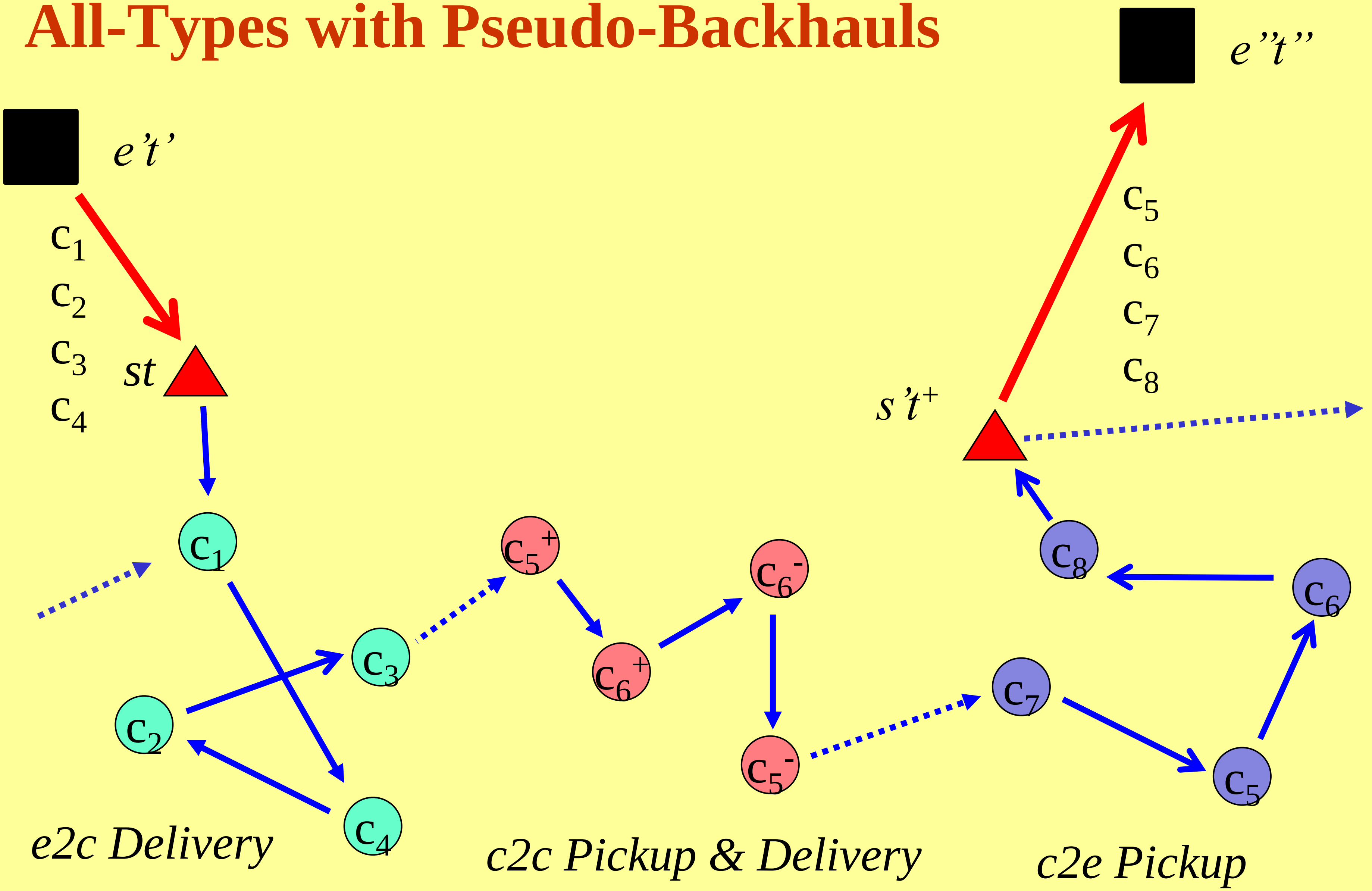
Authors	Goetschalckx and Jacobs-Blecha (1989)		Toth and Vigo (1997)	
	Cost	GAP to BKS (%)	Cost	GAP to BKS (%)
Osman and Wassan (2002)	291261.7	0.25	708.42	1.09
Brandao (2006)	291160.5	0.21	702.15	0.19
Wassan (2007)	290981.8	0.15	706.48	0.81
Our work	**290964.4**	**0.14**	**705.49**	**0.67**

Conclusions and Perspectives

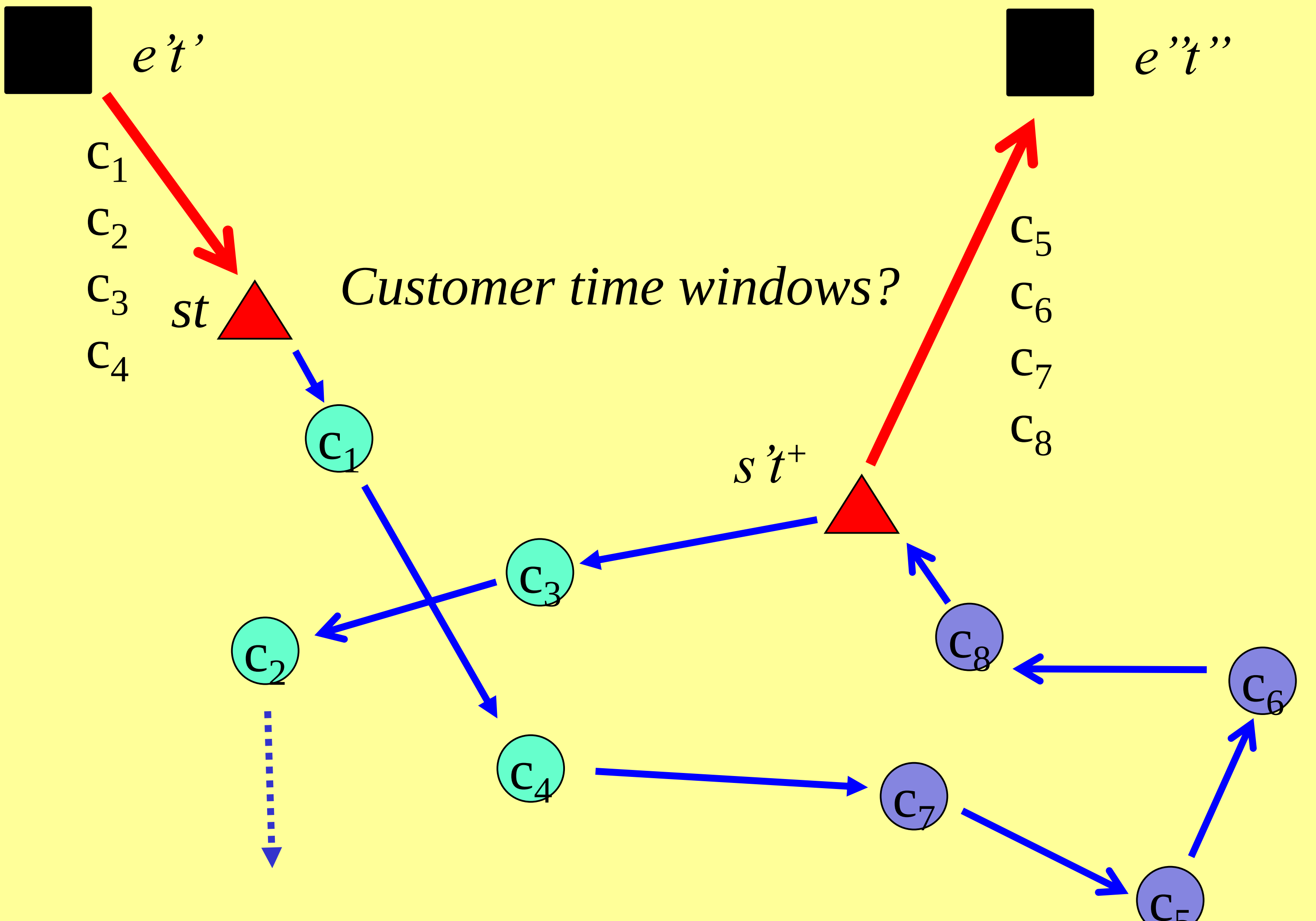
- The algorithm performs well on rather large instances
- It also performs well on well-known VRP with backhauls problems
- Interesting future questions
  - There are more neighborhoods one could try out
  - More complex operation strategies
  - Integrating c2c movements
  - Bounds and “exact” methods
  - Two-level settings



All-Types with Pseudo-Backhauls

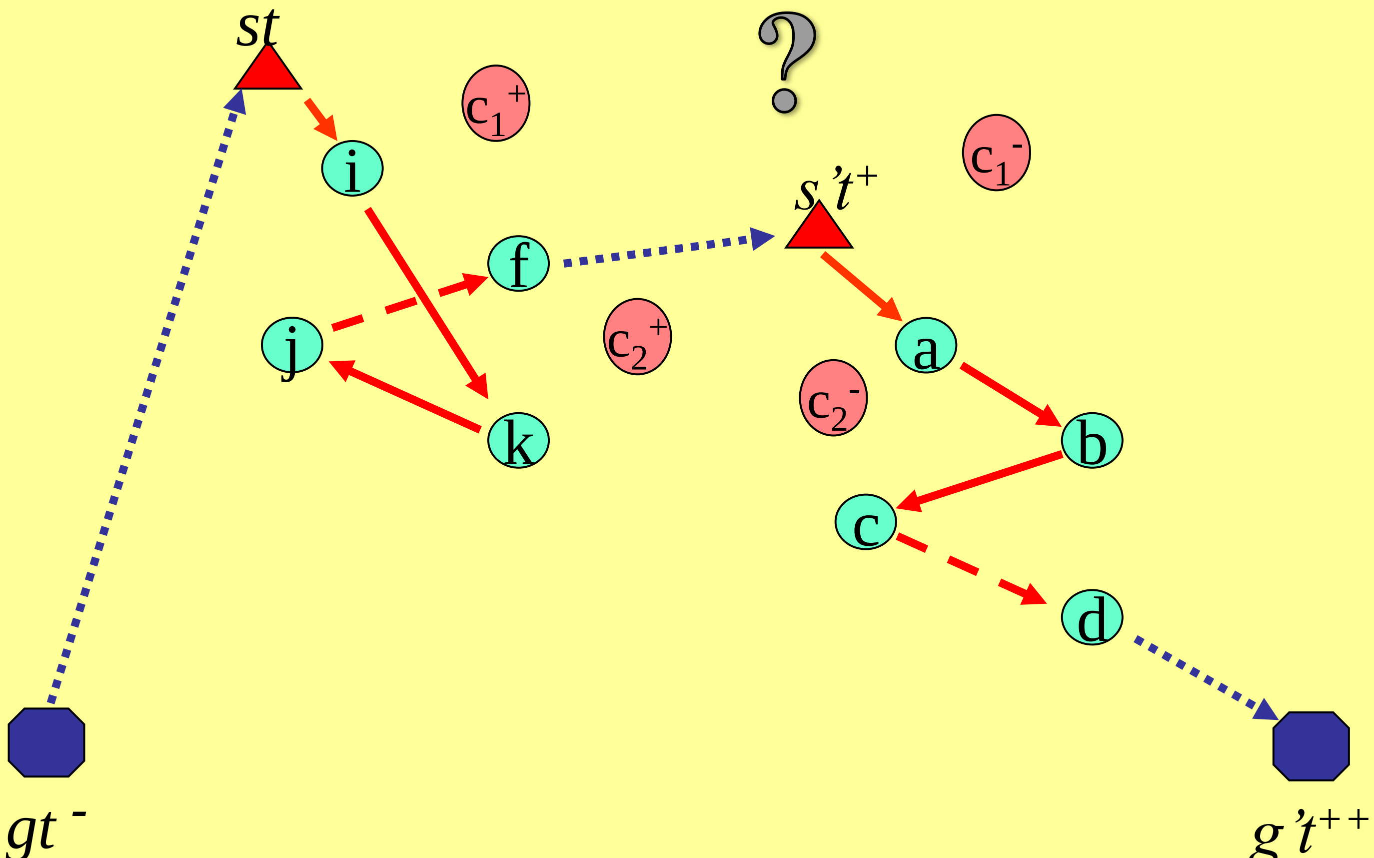


Interlacing Delivery & Pickup Phases in e2c+c2e

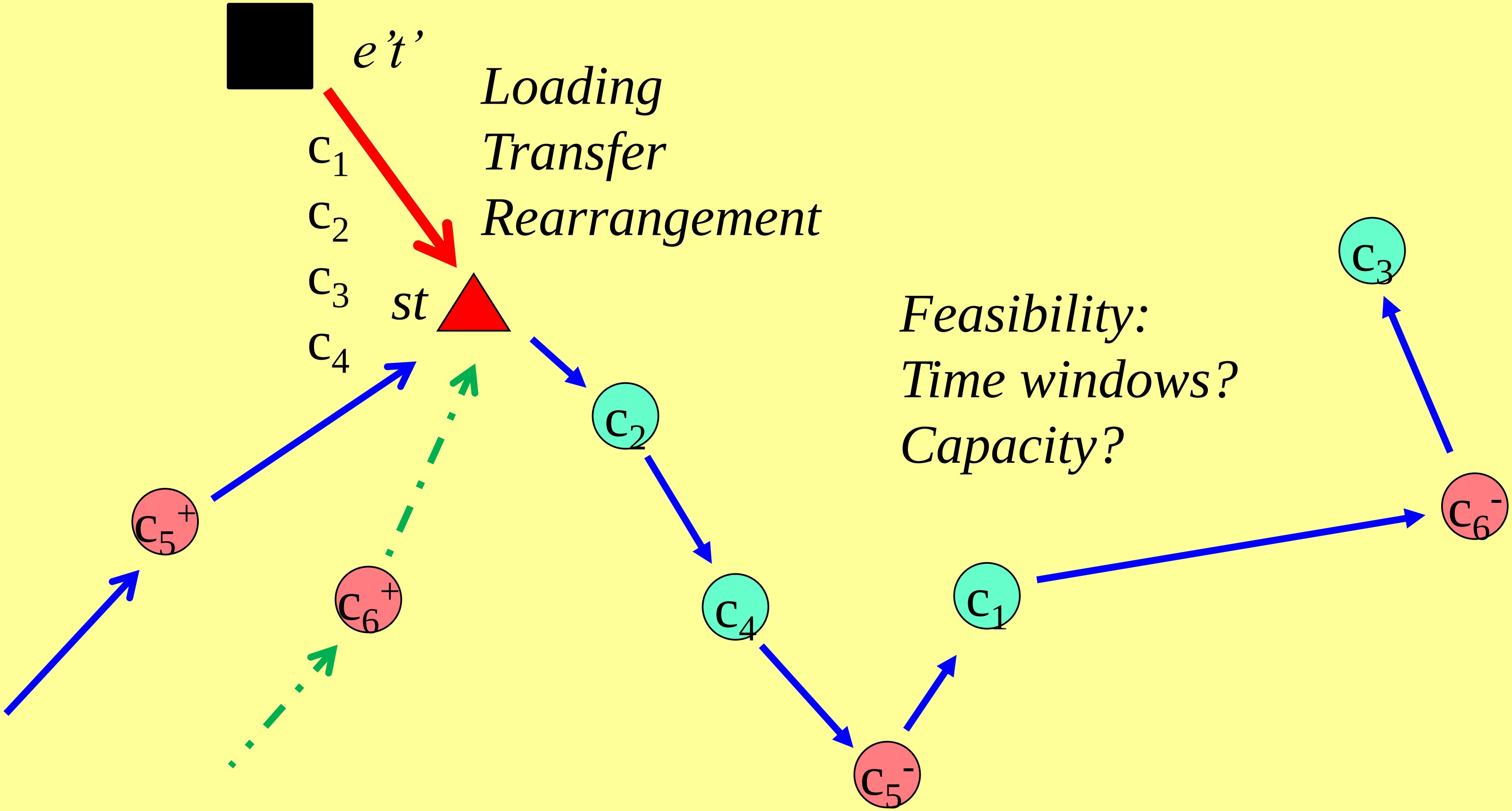


Integrating Intra-City Demand

- What operations allowed at supply points when considering c2c demand?



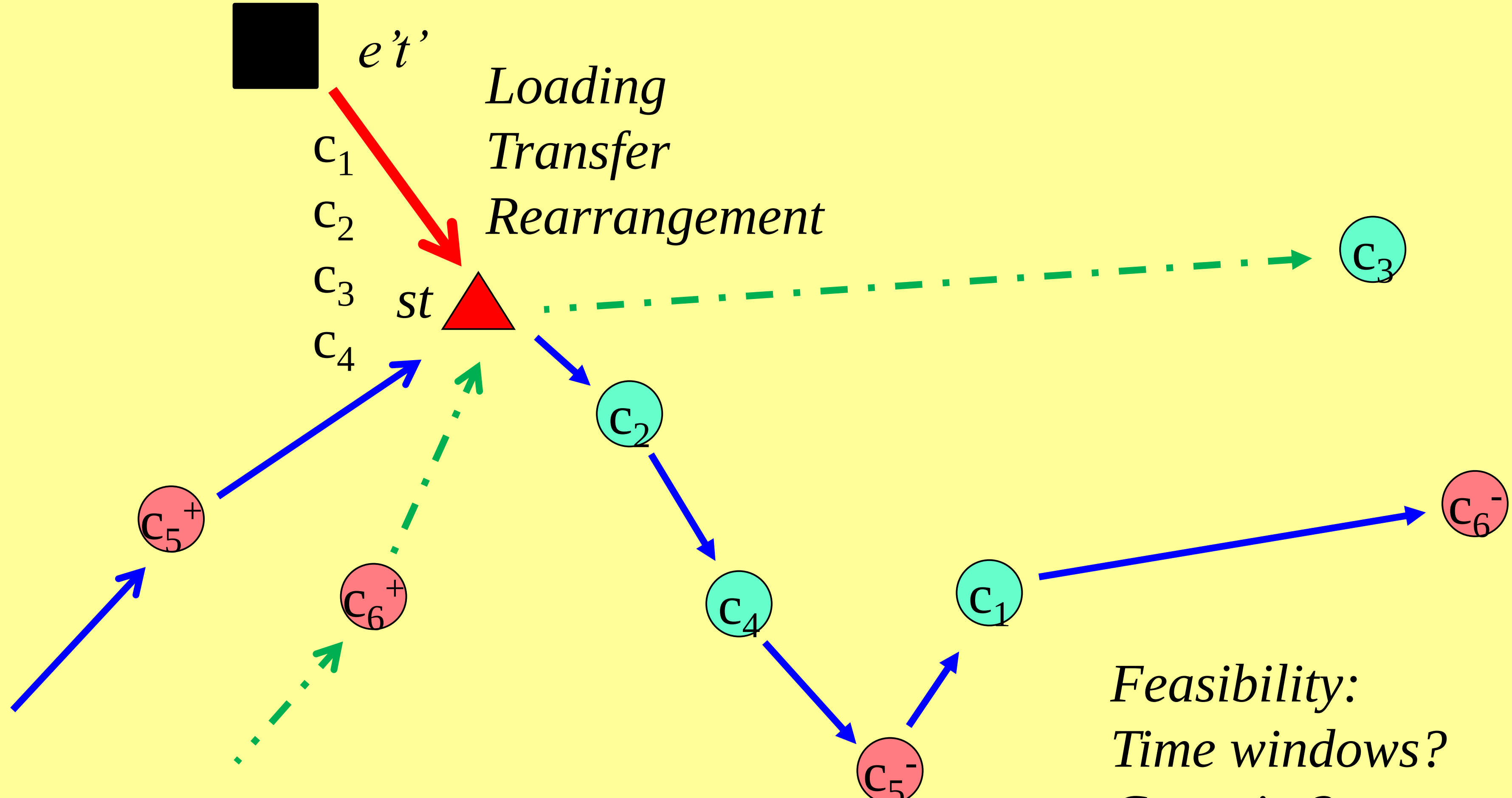
Joint e2c & c2c Routing – What Satellite Work?



Loading
Transfer
Rearrangement

Feasibility:
Time windows?
Capacity?

Joint 2c & c2c Routing – What Satellite Work?



Feasibility:
Time windows?
Capacity?

Perspectives

- The more comprehensive the integration & "complete" the system: the larger the benefits
  - Less vehicles, congestion, pollution, …
- The more flexibility is allowed in adjusting the plan to "revealed" demand: the larger the benefits
  - Costs, km traveled, capacity utilization …
- The more important the management challenges
  - Flexibility & agility work rules & labor relations
- The more "interesting" the methodological challenges