



Dynamic Vehicle Routing and Dispatching

Jean-Yves Potvin

Département d'informatique et recherche opérationnelle
and
Centre interuniversitaire de recherche sur les réseaux d'entreprise,
la logistique et le transport (CIRRELT)

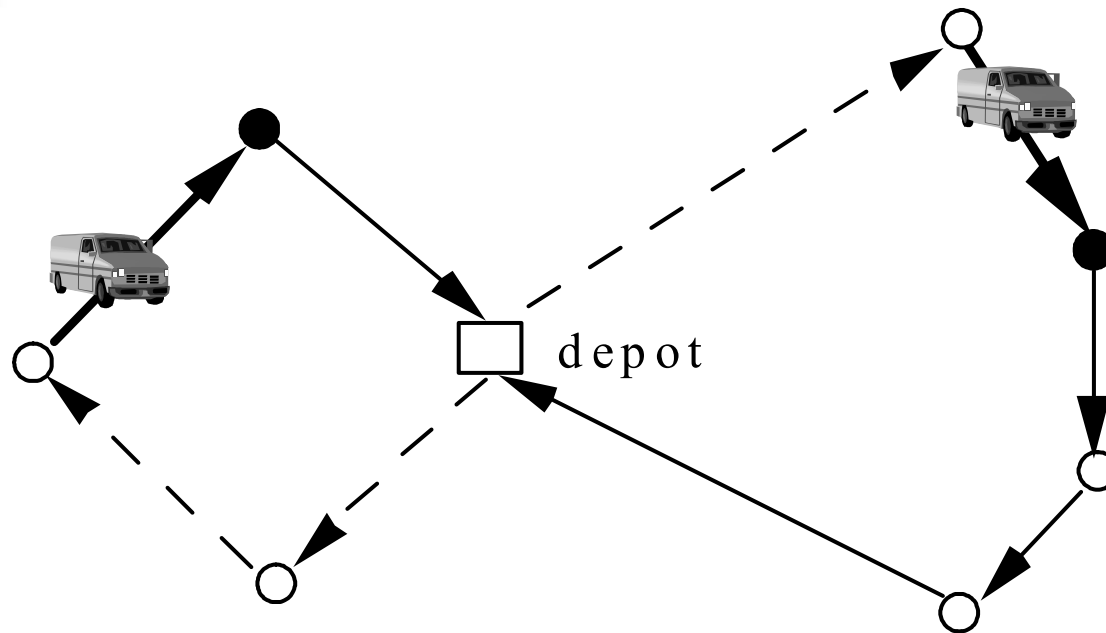
Université de Montréal



Outline

- Introduction
- Adaptation of « static » algorithms
- Diversion
- Exploitation of knowledge about the future
- Conclusion

Introduction



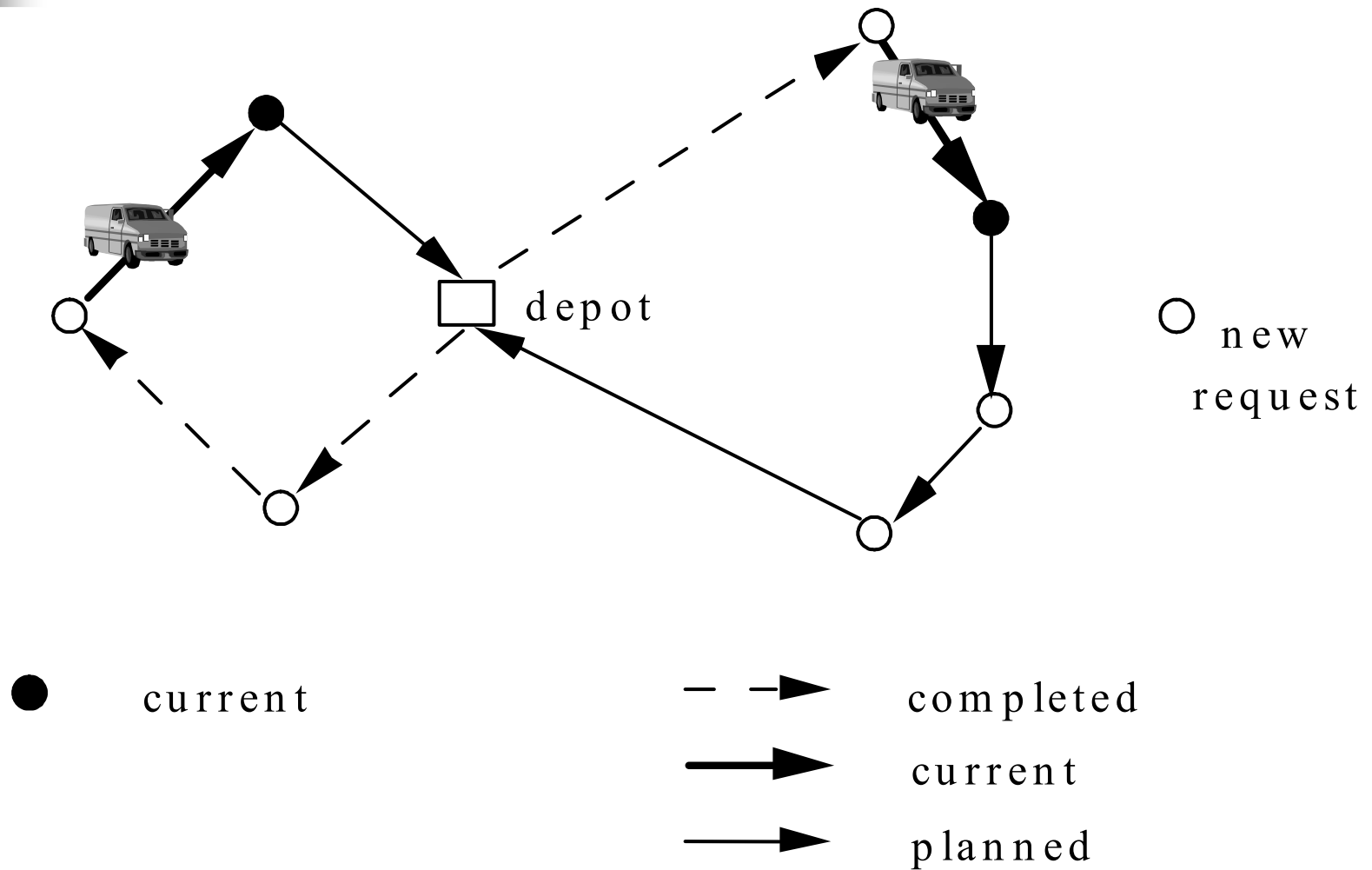
● current

- - ➔ completed

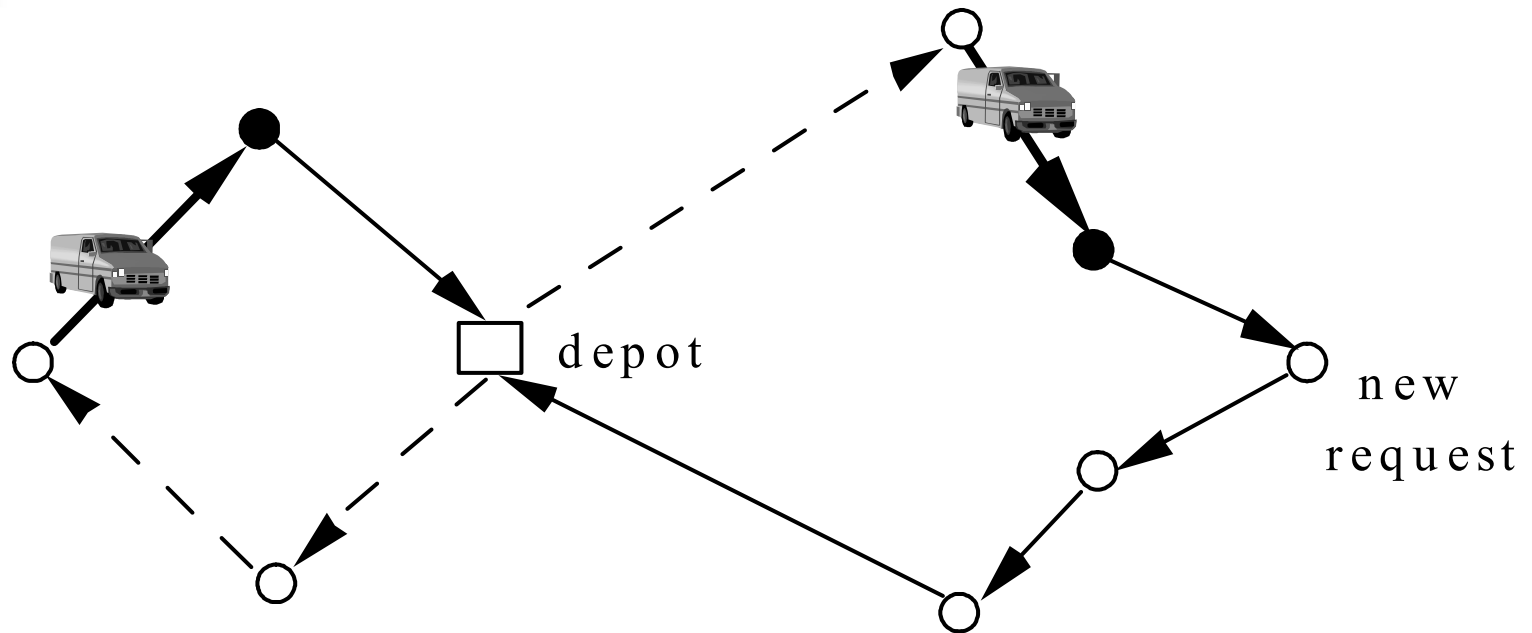
➔ current

➔ planned

Introduction



Introduction



● current

- - → completed

→ current

→ planned



Introduction

- New interest due to technological advances in:
 - Communication (e.g., GPS, mobile communication devices)
 - Information processing capabilities.



Introduction

- On-Line: new information is revealed as the routes are executed.
 - Customer requests
 - Travel times
 - Others
- Real-Time: on-line + a quick reaction is required.



Literature

- Gendreau M., J-Y. Potvin, "Dynamic Vehicle Routing and Dispatching", in Fleet Management and Logistics, T.G. Crainic and G. Laporte (eds.), Kluwer, 115-126, 1998.
- Ghiani G., F. Guerriero, G. Laporte, R. Musmanno, "Real-Time Vehicle Routing: Solution Concepts, Algorithms and Parallel Computing Strategies", EJOR 151, 1-11, 2003.
- Powell W.B., Jaillet P., Odoni A., " Stochastic and Dynamic Networks and Routing " in Network Routing, Handbooks in Operations Research and Management Science 8, M.O. Ball et al. (eds.), 141-295, Elsevier, 1995.



Literature

- Psaraftis H.N., "Dynamic Vehicle Routing: Status and Prospects", *Annals of Operations Research* 61, 143-164, 1995.
- Psaraftis H.N., "Dynamic Vehicle Routing Problems", in *Vehicle Routing: Methods and Studies*, B.L. Golden and A.A. Assad (eds.), North-Holland, 223-248, 1988.
- Savelsbergh M.W.P., Sol M., "DRIVE: Dynamic Routing of Independent VEHicles", *Operations Research* 46, 474-490, 1998.
- Séguin R., J.-Y. Potvin, M. Gendreau, T.G. Crainic and P. Marcotte, "Real-Time Decision Problems: An Operational Research Perspective", *Journal of the Operational Research Society* 48, 162-174, 1997.



Scope (with application examples)

	Local Area	Wide Area
Consolidation	Courier Dial-a-Ride	LTL Trucking
No Consolidation	Emergency services	TL Trucking



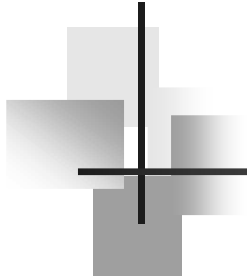
Scope (with application examples)

	Many-to-many	Many-to-one
Capacity	Dial-a-Ride	Feeder systems
No capacity	Urban courier services	International courier (local routes)



Scope

- Dynamic events
 - new customer requests.



Adaptation of « static » algorithms



Adaptation of static algorithms

- Each new request generates a new static problem.
- Thus, a series of static problems are defined and solved as time unfolds.
- Each static problem includes only requests that are known with certainty; no integration of future forecasted requests (myopic).



Adaptation of static algorithms

- *Gendreau M., F. Guertin, J.-Y. Potvin and É. Taillard, "Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching", *Transportation Science* 33, 381-390, 1999.
- *Gendreau M., F. Guertin, J.-Y. Potvin, R. Séguin, "Neighborhood Search Heuristics for a Dynamic Vehicle Dispatching Problem with Pick-Ups and Deliveries", forthcoming in *Transportation Research C*.
- Attanasio A., Cordeau J.-F., Ghiani G., Laporte G., "Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem", *Parallel Computing* 30, 377-387, 2004.
- Montemanni R., L.M. Gambardella, A.E. Rizzoli, A.V. Donati, "A new algorithm for a Dynamic Vehicle Routing Problem based on Ant Colony System", Technical Report IDSIA-23-02, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, Lugano, Switzerland, 2002.

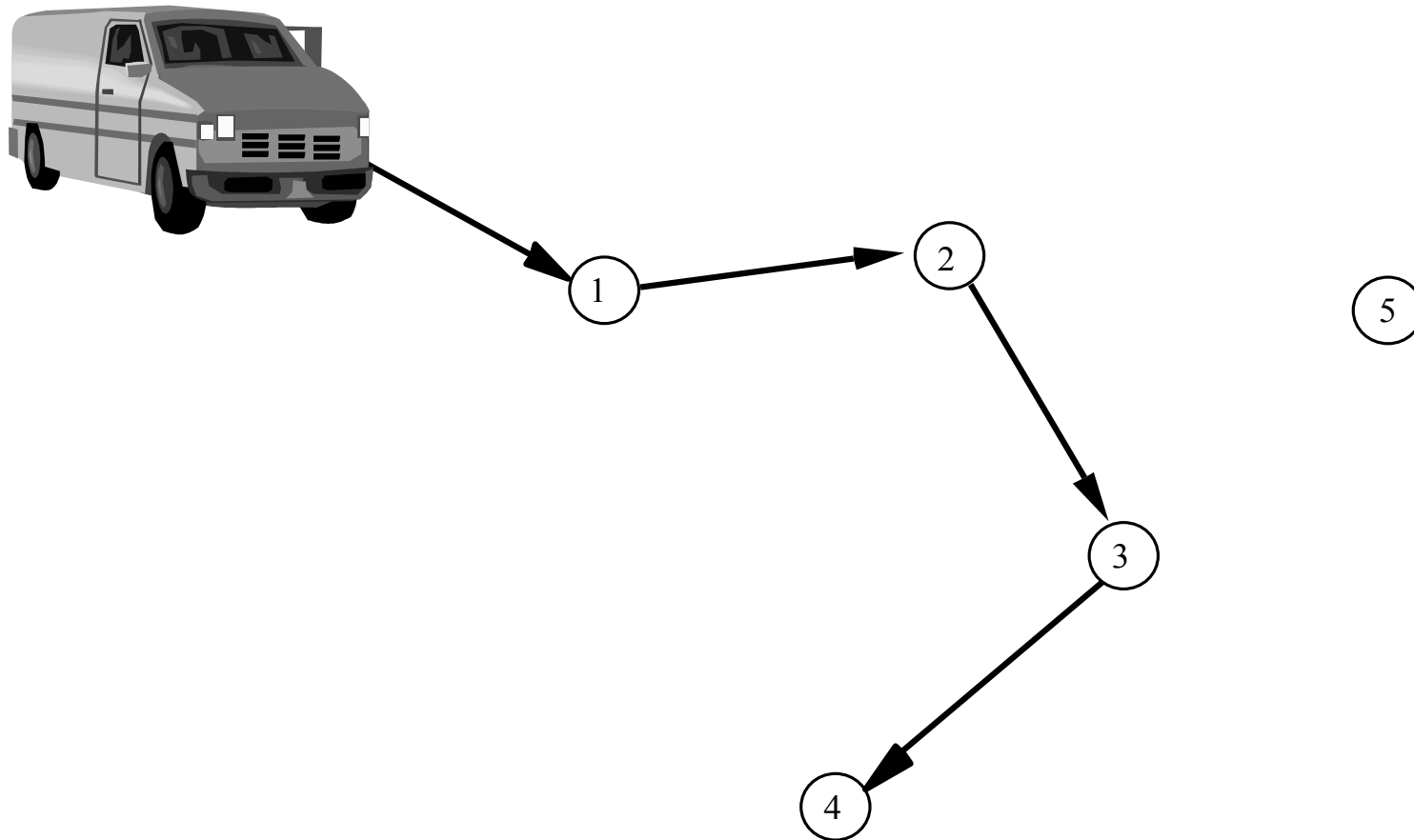


Problem definition

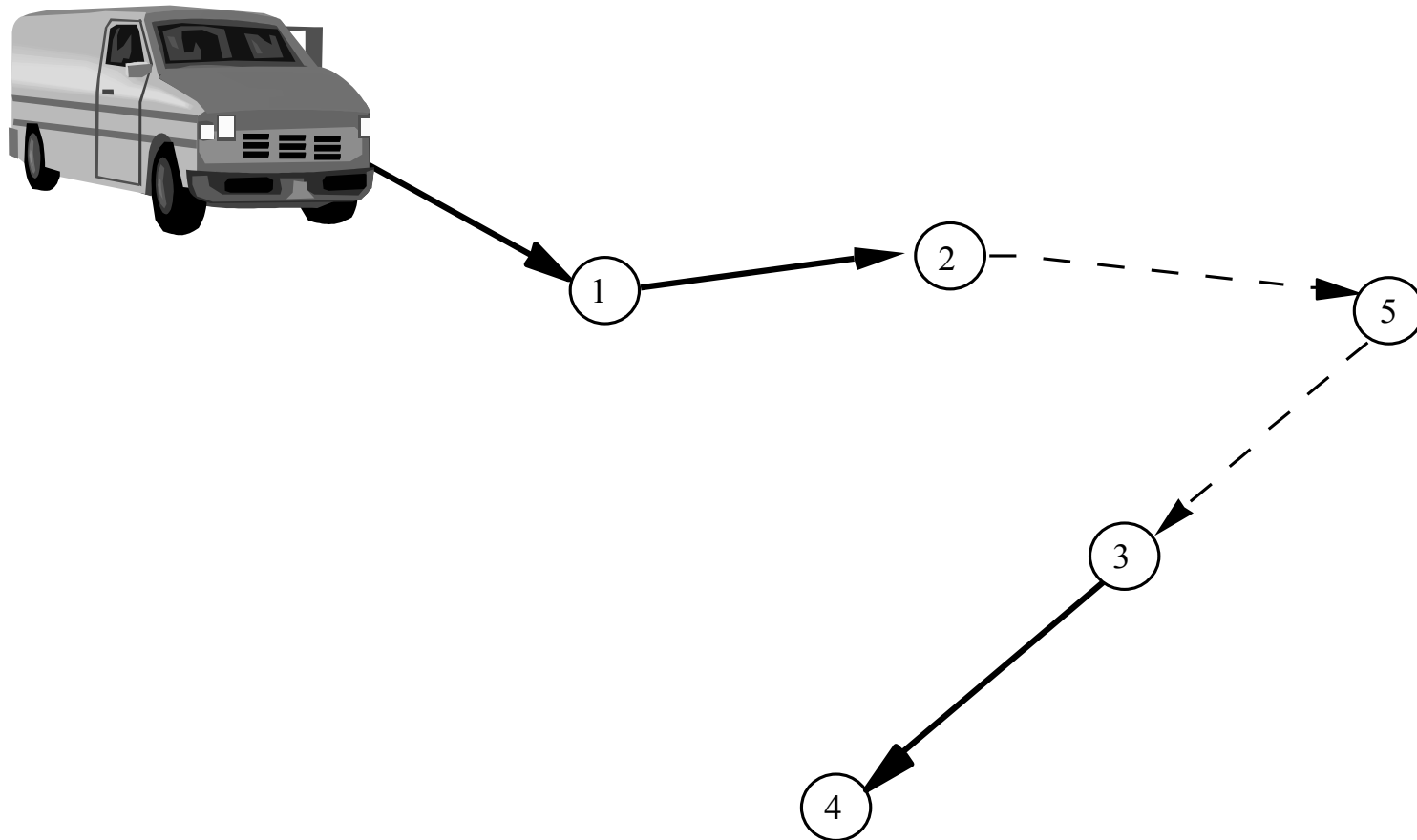
- A fleet of vehicles.
- A planned route for each vehicle.
- Each new request must be assigned to a vehicle and inserted in its planned route.

- Travel times are known with certainty.
- Objective function.
 - Minimize operations costs
 - Maximize customer satisfaction

Many-to-One



Many-to-One





Problem definition

- Each customer request is characterized by:
 - time of occurrence
 - single location (pick-up)
 - service time or dwell time
 - soft time window



Problem definition

- Fleet of vehicles
 - fixed number
 - identical
 - unlimited capacity
- Depot
 - each vehicle route starts and ends at the depot
 - hard time deadline



Problem definition

- Travel times are deterministic
- Objective
 - minimize total travel time plus sum of lateness at customer locations.



Operations mode

- Customers
 - A number of customer requests might be known in advance (partially dynamic).
 - Customers call the central dispatch office for a (single) transportation request.



Operations mode

- Dispatch office
 - When a new customer calls in, the current destination and planned route of every vehicle is known.
 - Task: insert the new request into one of the planned routes at minimum cost.



Operations mode

■ Drivers

- Communication with dispatch office takes place at customer locations only.
- The drivers call the dispatch office when they are ready to depart from their current location (to know their next destination).
- The drivers do not know their planned route.
- When a vehicle is in movement, the current destination is fixed.



Problem-solving approach

- Adaptation of an algorithm developed for the static version of the problem (VRPSTW).
- A static problem is defined and solved each time an input update occurs.
- The static problem is defined over known requests only.
- The algorithm is a tabu search heuristic; it runs between the occurrence of events.



Events

- New customer request
 - insert at least cost in planned routes
 - improve with tabu search
- End of service
 - indicate next destination to driver (based on best known solution)
 - update planned routes
 - improve with tabu search

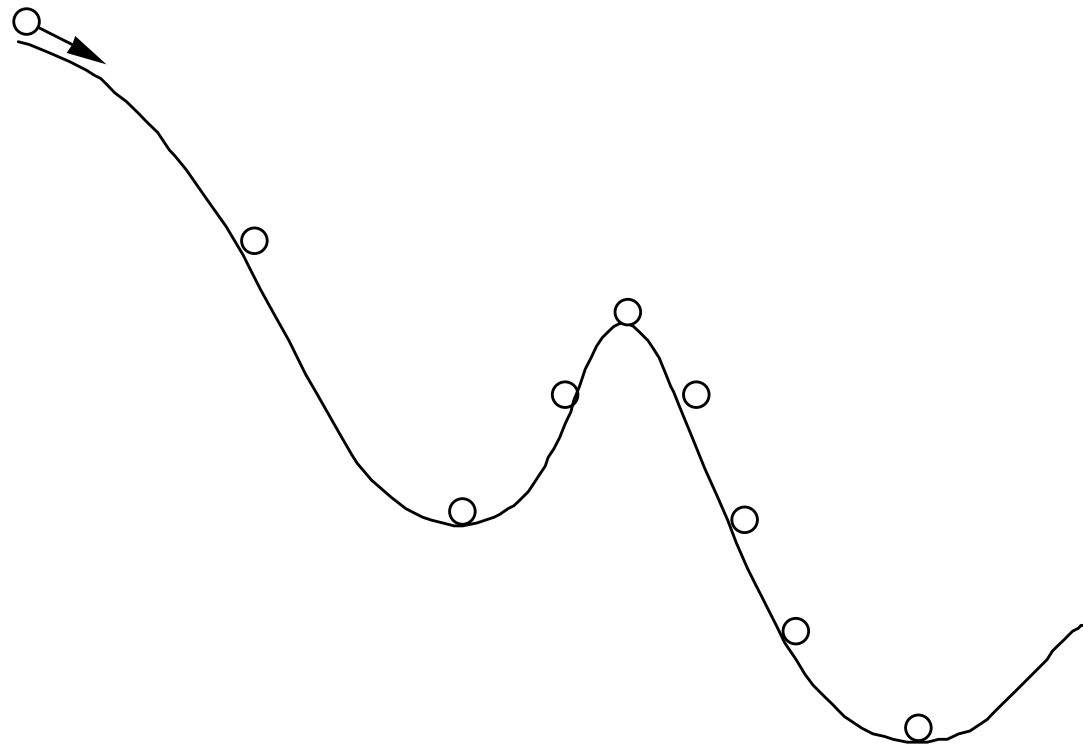


Tabu search

0. Generate an initial solution (current solution).
1. Repeat until a stopping criterion is met:
 - 1.1 Apply a predefined operator on the current solution; this operator defines a class of modifications that lead to multiple neighbor solutions.
 - 1.2 Select the best (non tabu) solution in this neighborhood of solutions and set it as the new current solution.
2. Return the best solution found.

Interesting property for dynamic problems:
Anytime Property

Tabu search

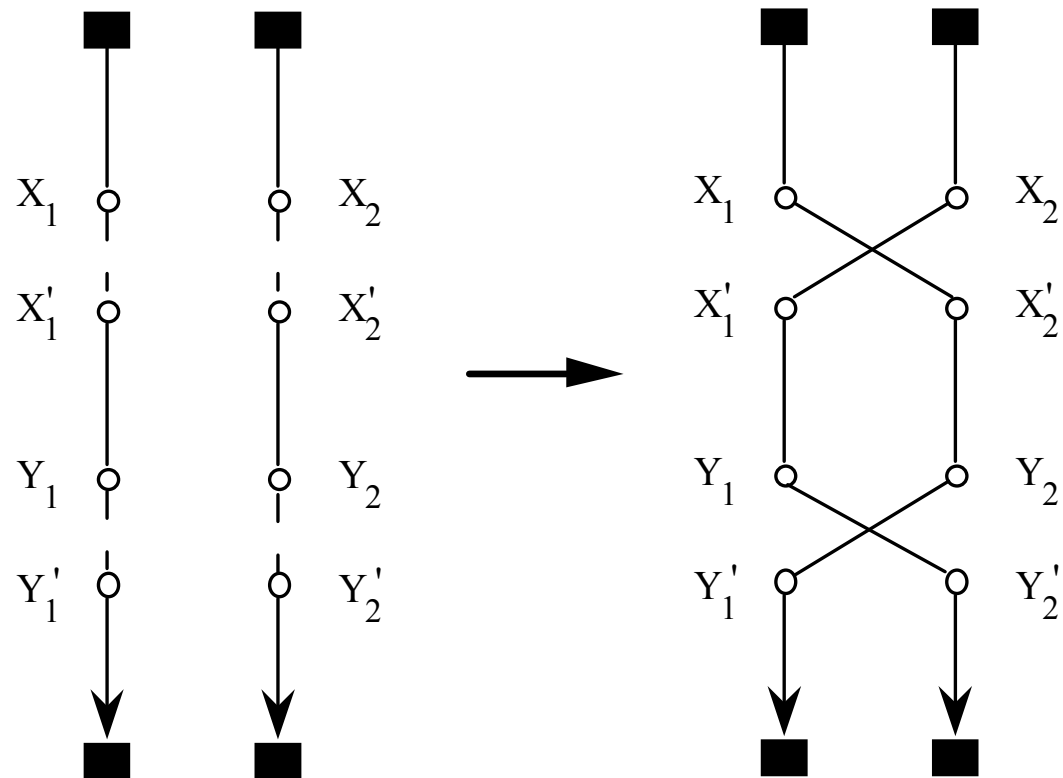




Tabu search

- Neighborhood based on CROSS exchanges.
- Neighborhood reduction procedures.
- Approximations are used to speed up the evaluation of each move.

CROSS exchanges





Tabu search

- Neighborhood based on CROSS exchanges.
- Neighborhood reduction procedures.
- Approximations are used to speed up the evaluation of each move.



The overall algorithm

- Repeat until next event:
 - fetch an initial solution from the **adaptive memory**
 - for a number of cycles do:
 - **decompose** the current set of routes into a number of subsets of routes
 - apply *Tabu Search* to each subset of routes
 - store the best solution found in adaptive memory (if indicated)



Adaptive memory

- Pool of solutions (routes) made of best previously visited solutions.



Adaptive memory

- Fetch:
 - New initial solutions for tabu search are created by selecting and combining routes associated with different solutions stored in this memory.
- Store:
 - A solution produced by tabu search is stored in adaptive memory if :
 - the memory is not filled yet
 - OR
 - it is better than the worst solution in memory.



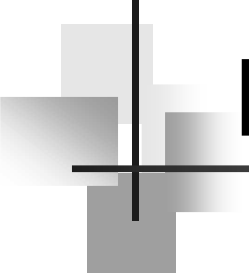
Multiple Plan Approach (MPA)

- Maintain multiple routing plans that are consistent with the current state of information.
- Bent R. and P. Van Hentenryck, "Scenario-Based Planning for Partially Dynamic Vehicle Routing with Stochastic Customers", *Operations Research* 52, 977-987, 2004.



Multiple Plan Approach (MPA)

- A “distinguished” plan or solution is followed until the occurrence of the next event.
- In the mean time, better planned routes over known requests are looked for by the optimization algorithm



Events (revisited)

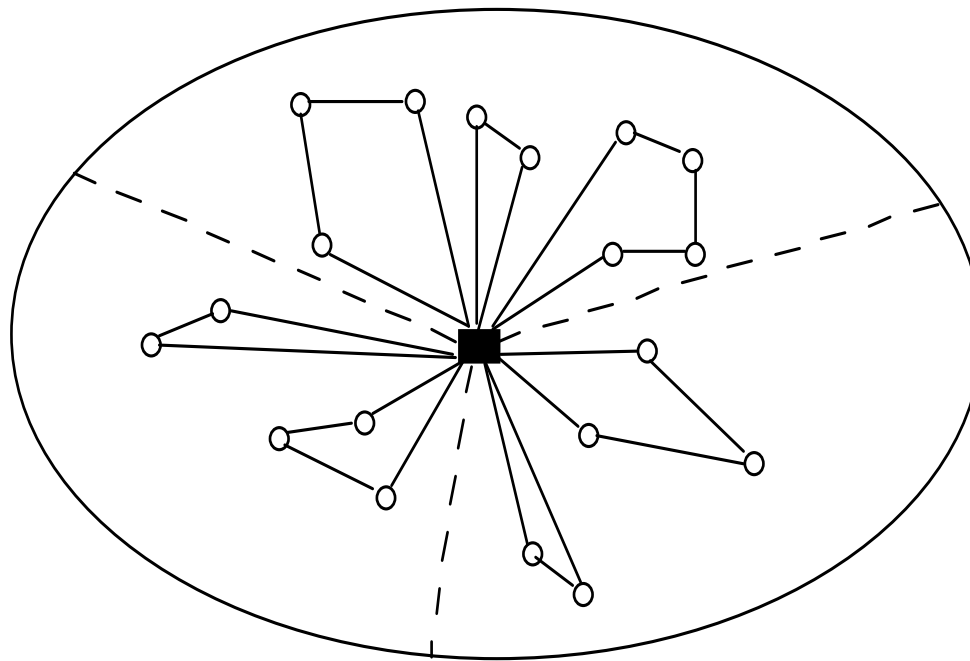
- New request
 - insert at least cost in each solution in adaptive memory
 - restart tabu search
- End of service
 - indicate next destination to driver based on best solution in adaptive memory
 - update other solutions in adaptive memory accordingly
 - restart tabu search



The overall algorithm

- Repeat until next event:
 - fetch an initial solution from the adaptive memory
 - for a number of cycles do:
 - **decompose** the current solution into a number of subsets of routes
 - apply *Tabu Search* to each subset of routes
 - store the best solution found in adaptive memory (if indicated)

Decomposition





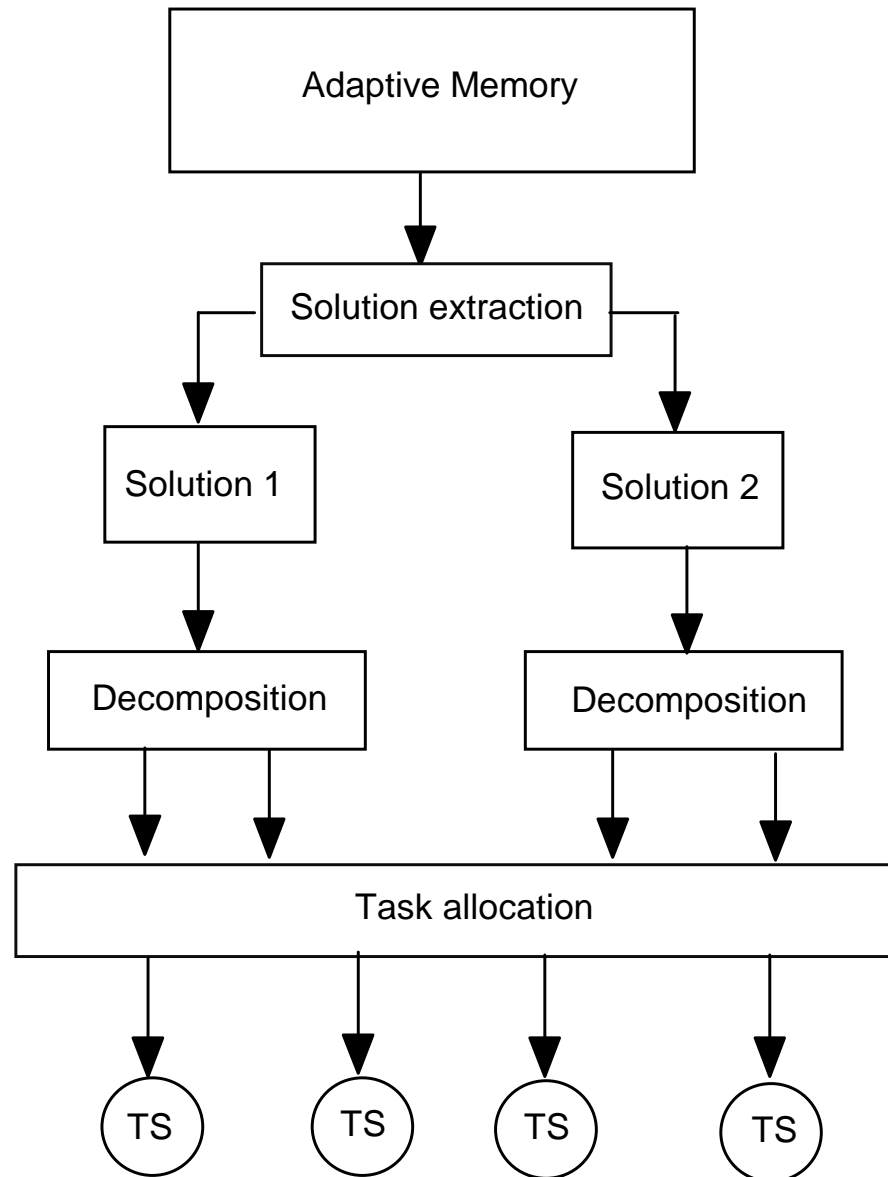
Parallel Implementation

- Provide a means to do more optimization work between events.
- Coarse-grained architecture
 - network of workstations
- Master-slave scheme where the master controls the slaves.
- Each slave runs a tabu search.



Parallel Implementation

- Two-level parallel scheme:
 - Many search threads run in parallel. These search threads communicate through the adaptive memory.
 - Within each search thread, the subproblems produced through the decomposition procedure are solved in parallel.





Computational results

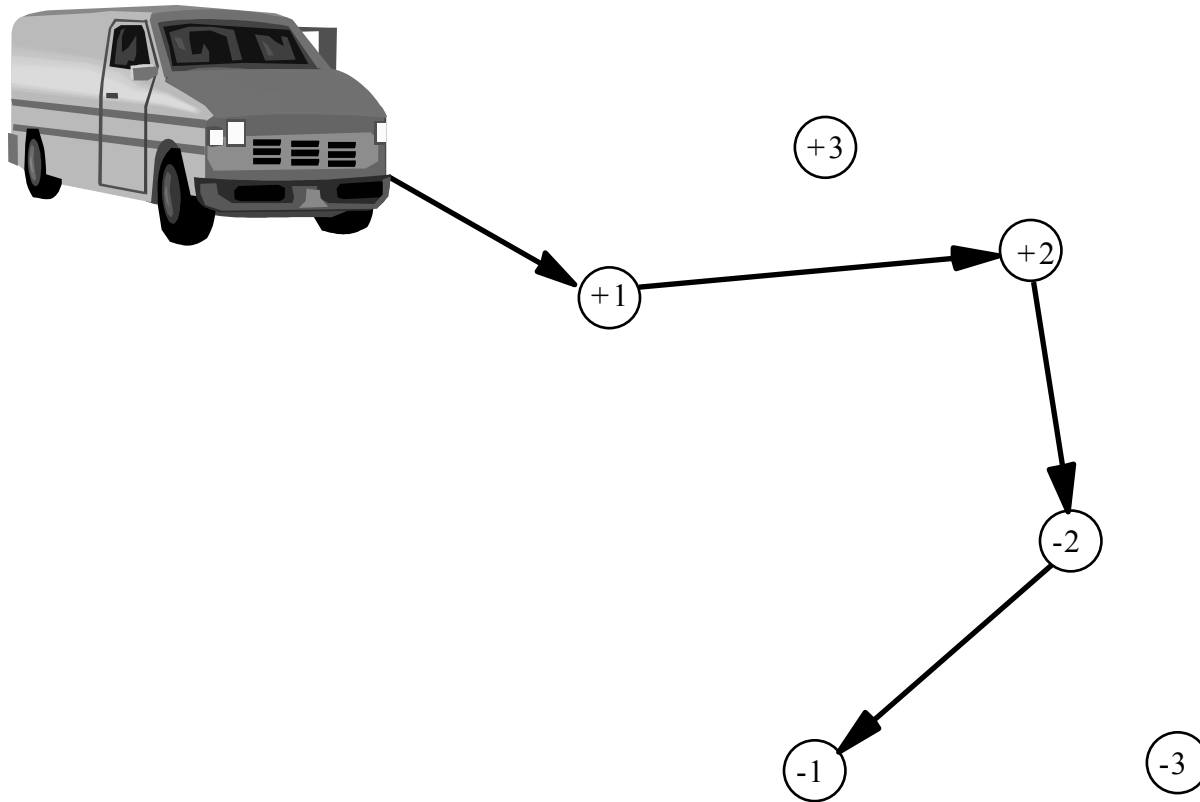
- Problem instances based on Solomon's benchmark VRPTWs.
- Scenarios with approximately 1 request per minute and 3 requests per minute.
- Sophisticated optimization procedures are beneficial over simple ones
 - number of unserved customers
 - total travel time and lateness
- Adaptive descent is competitive.



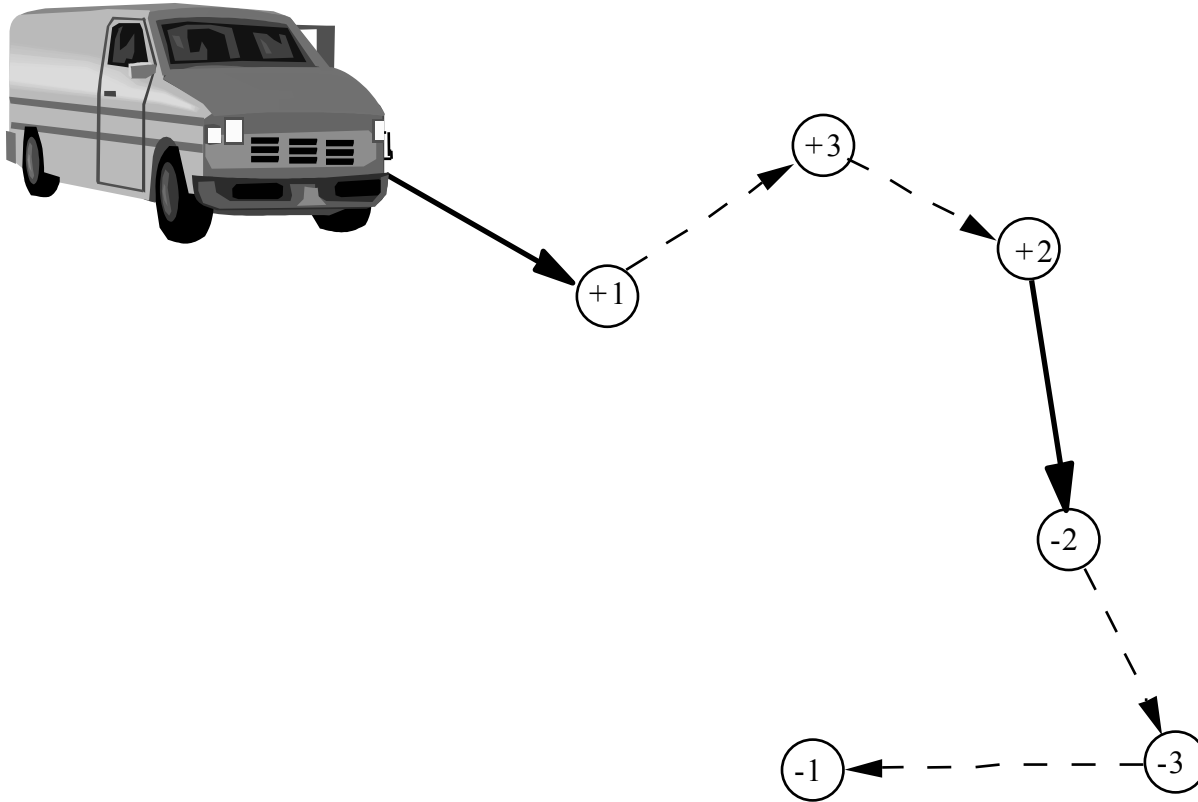
Extension to Many-to-Many

- Motivated by urban courier services.
- Each customer request has both a pick-up AND a delivery location (with precedence constraint).

Many-to-Many



Many-to-Many

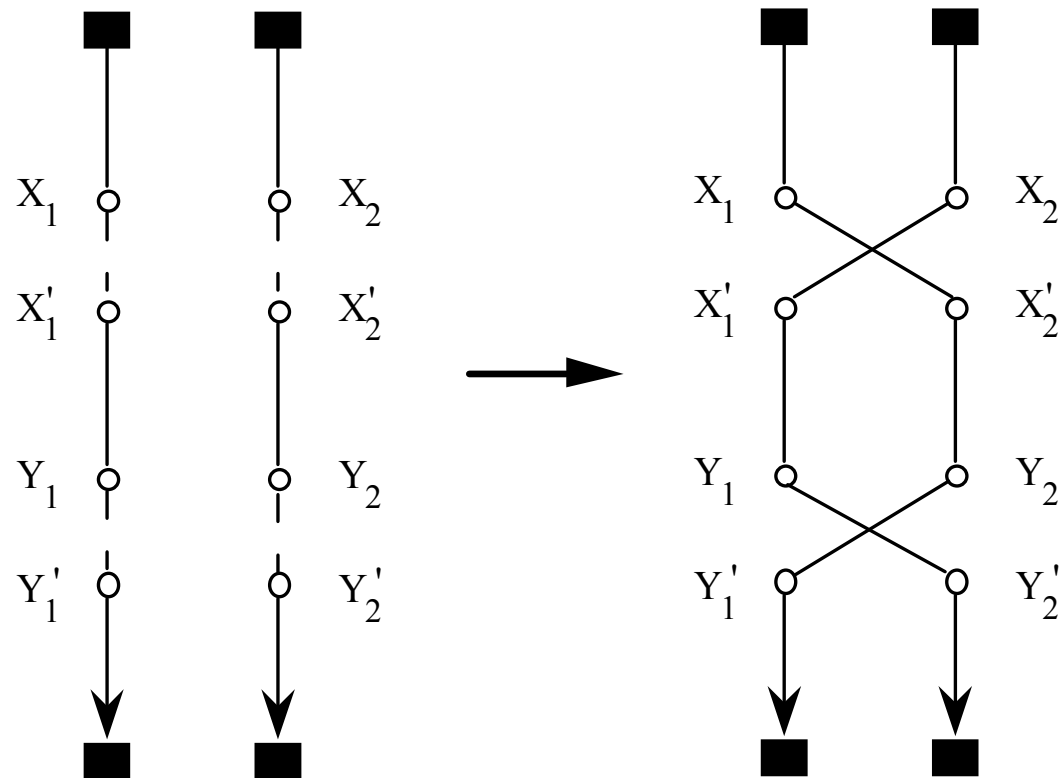




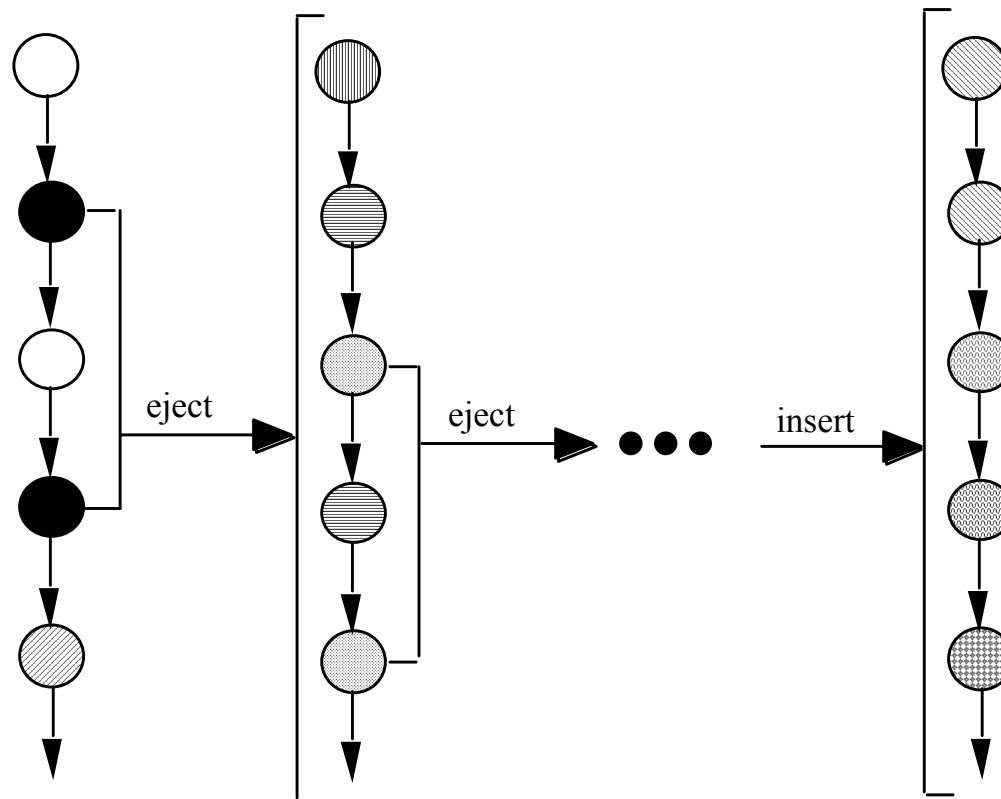
Extension to Many-to-Many

- CROSS exchanges are not appropriate.
- Ejection chains.

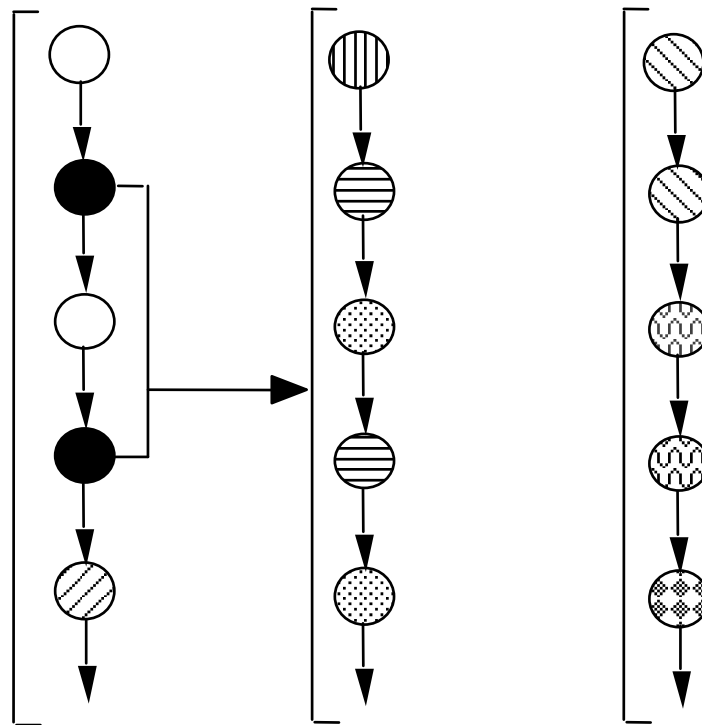
CROSS exchanges



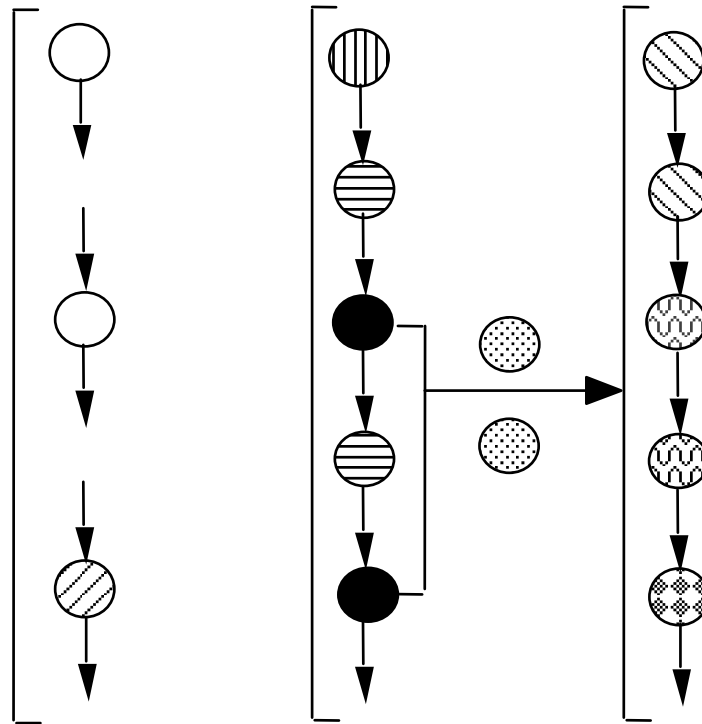
Ejection chains



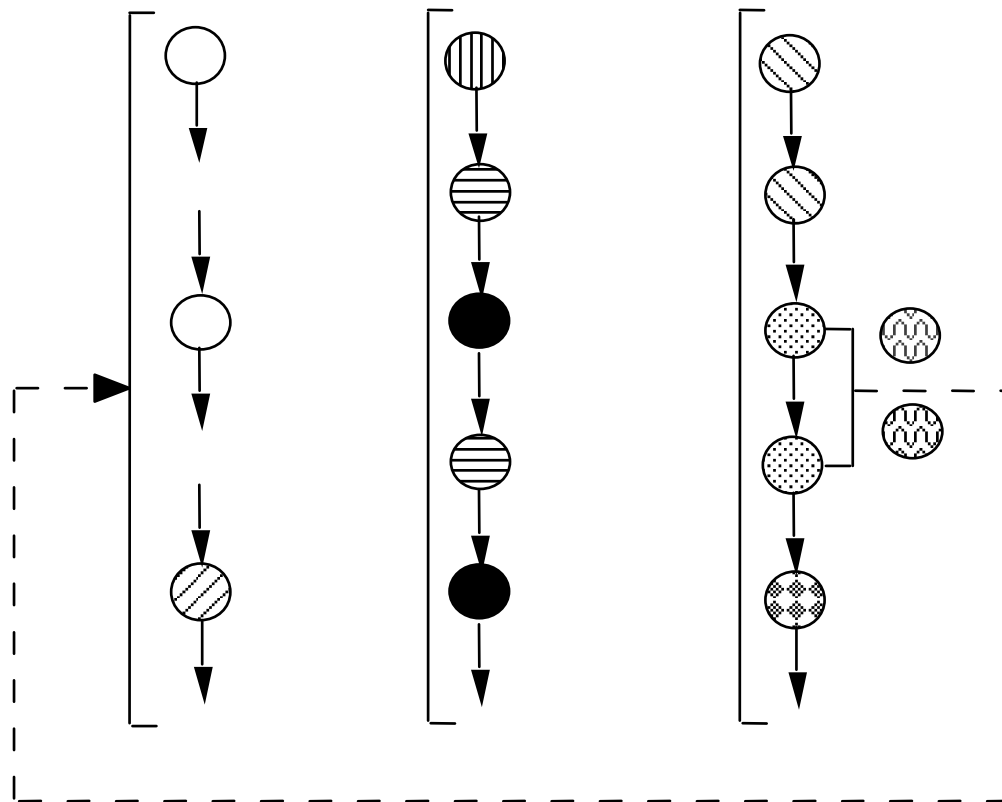
Ejection chains



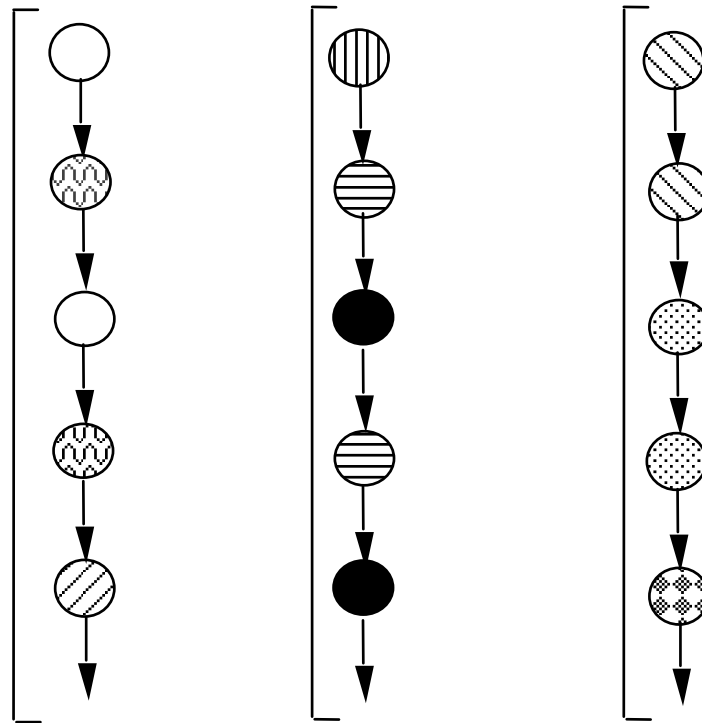
Ejection chains



Ejection chains



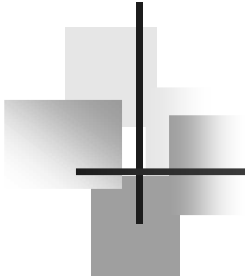
Ejection chains





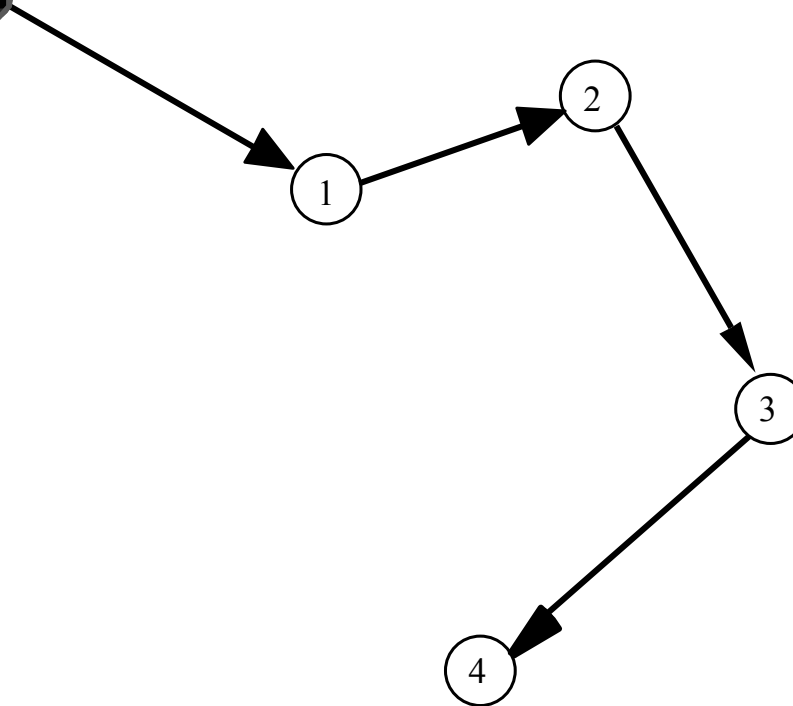
Rolling Horizon

- Often, planned routes are generated over some time horizon.
- This horizon moves with the current time.



Diversion

Diversion



Diversion



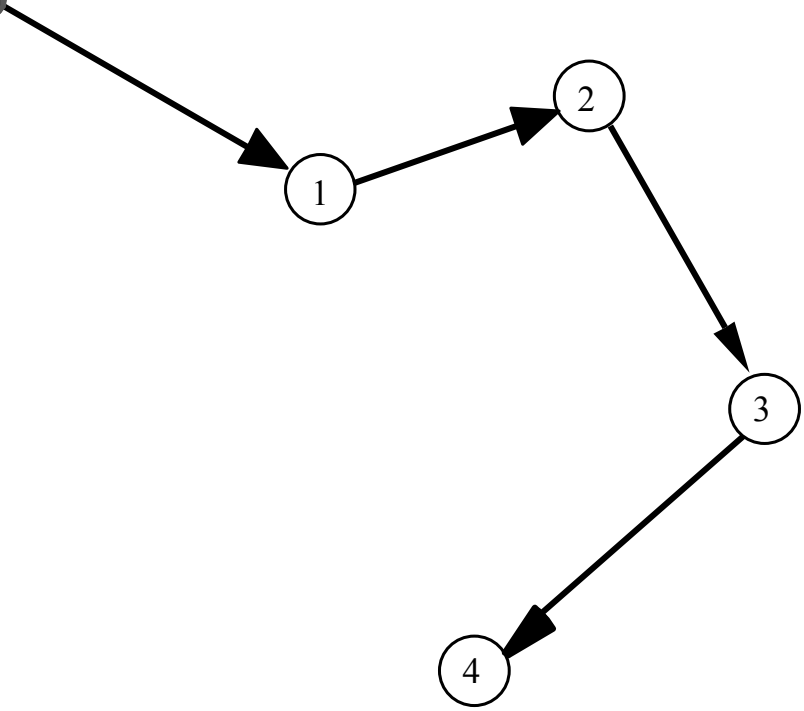
5

1

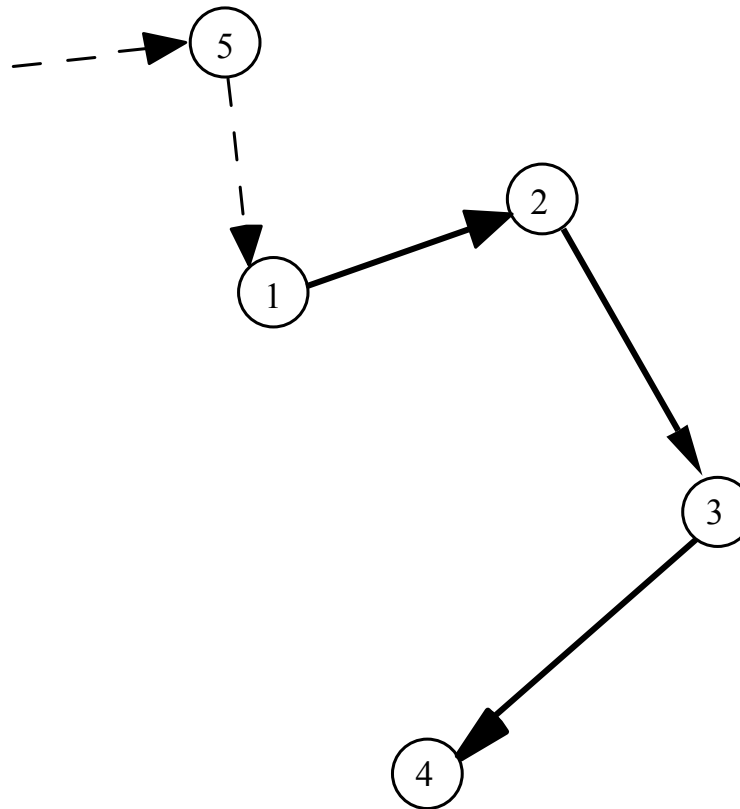
2

3

4



Diversion





Diversion

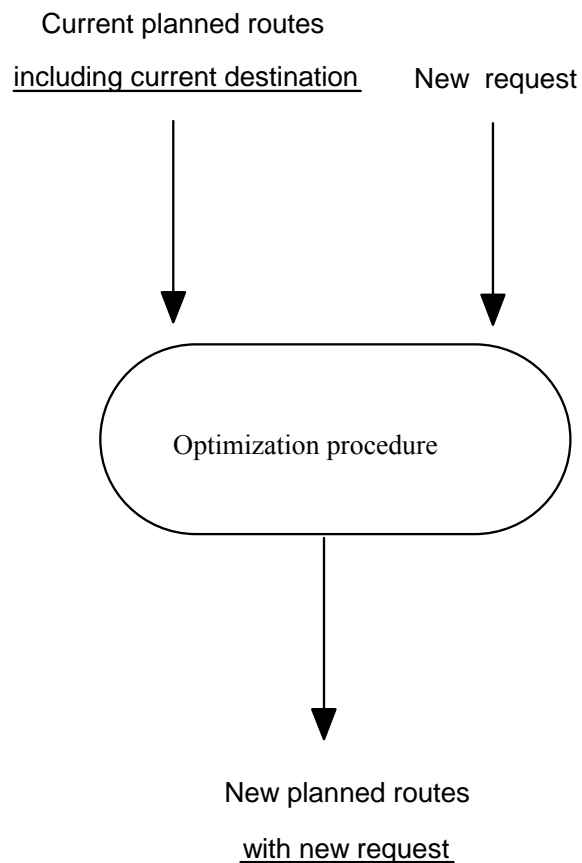
- Diversion means redirecting a vehicle to service a new request close to its current position.
- Implies a more sophisticated communication scheme between the drivers and the central dispatch office.



Diversion

- *Ichoua S., Gendreau M., Potvin J.-Y., "Diversion Issues in Real-Time Vehicle Dispatching", *Transportation Science* 34, 426-438, 2000.
- Regan A.C., Mahmassani, H.S., Jaillet, P., "Improving Efficiency of Commercial Vehicle Operations Using Real-time Information: Potential Uses and Assignment Strategies," *Transportation Research Record* 1493, 188-197, 1995.

Diversion



- At the end of optimization, classical diversion is only one possible outcome.
- In our case, the optimization procedure is a tabu search with adaptive memory.



Diversion

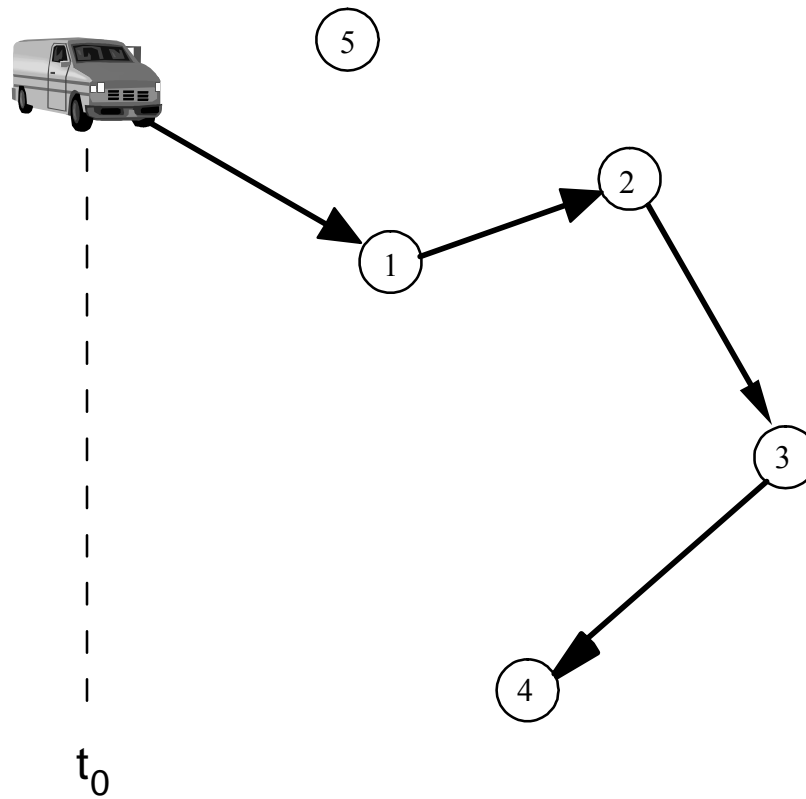
- Time pressure is important (vehicles are moving fast).
- Time “contract” for optimization
 - during time contract, follow best known solution;
 - at the end of time contract, return best solution found and implement it.



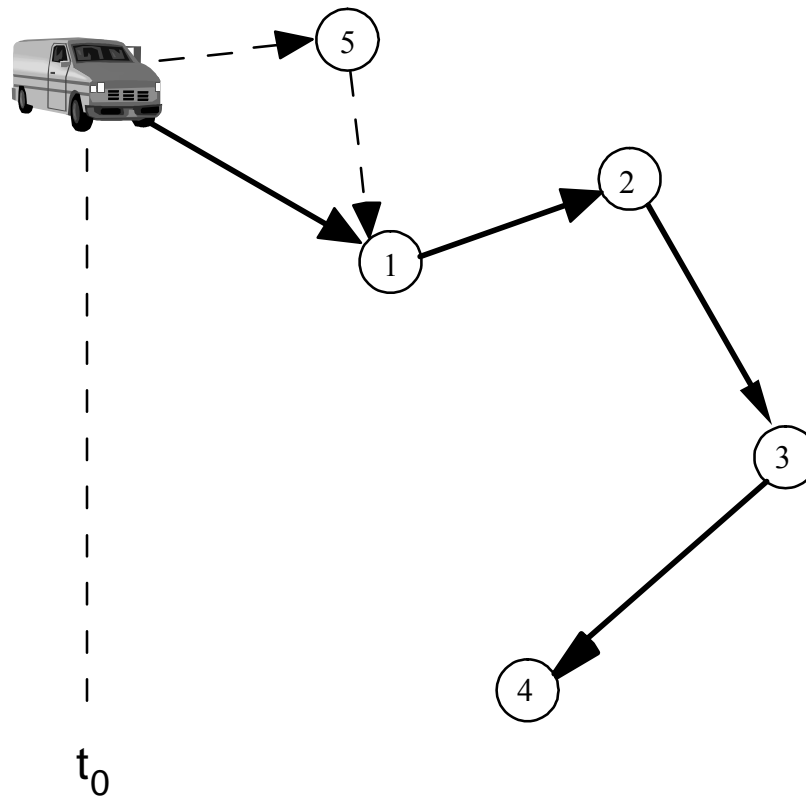
Time contract

- Solution quality/CPU time trade-off:
 - if CPU time is too small: poor optimization;
 - If CPU time is too large: diversion opportunities are lost.

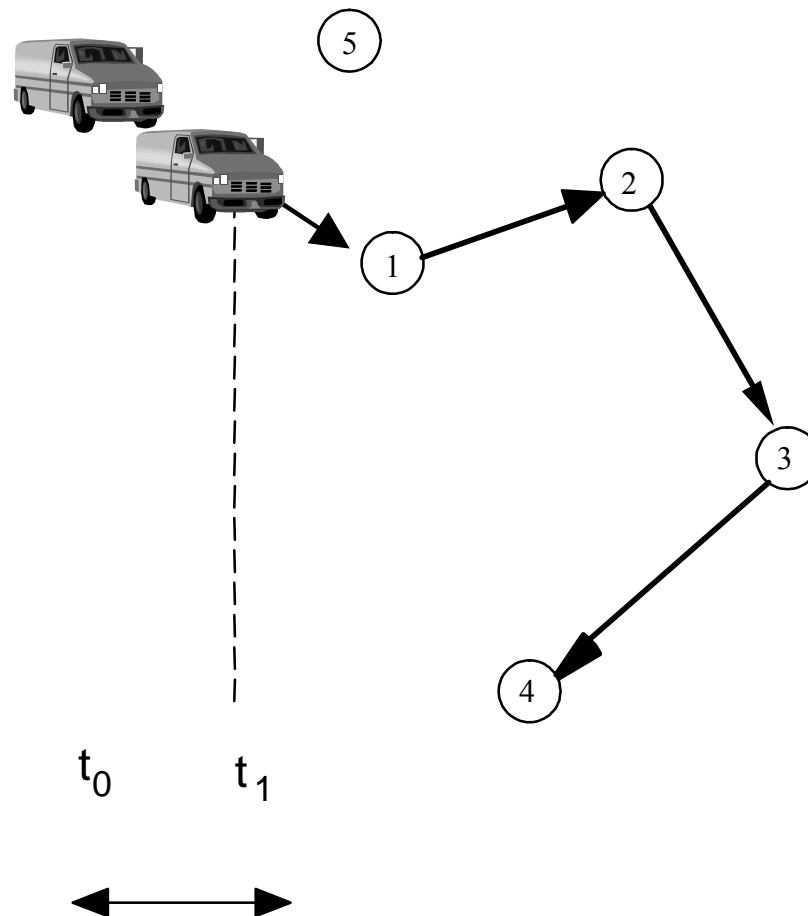
Time contract



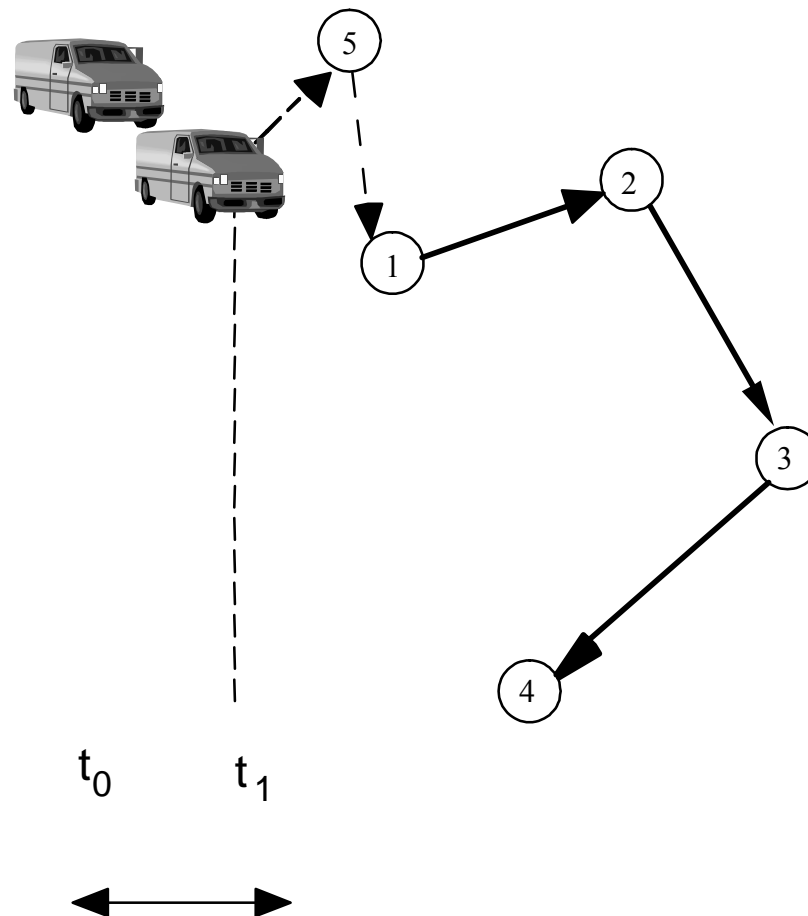
Time contract



Time contract



Time contract





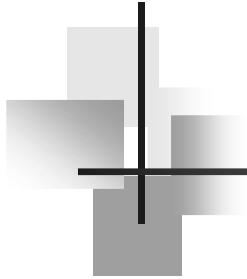
Time projection

- Before applying the optimization procedure, you first need to perform a time projection based on:
 - Best known solution
 - Duration of time contract
- The solution returned by the optimization procedure can then be directly implemented



Time contract

- Duration of time contract: moving average of inter-arrival times of a number of previous requests.
- Time contracts vary between 4 and 7 seconds, for scenarios with 3 to 5 requests per minute.
- Results show the benefits obtained by considering diversion, with regard to:
 - number of unserved customers;
 - total travel time and lateness (1% to 4%).



Exploiting knowledge about
the future



Exploiting knowledge about the future

Knowledge of probability distribution in space and time of future incoming requests can be exploited to generate more robust solutions.



Multiple Scenario Approach

- Bent R. and P. Van Hentenryck, "Scenario Based Planning for Partially Dynamic Vehicle Routing with Stochastic Customers", *Operations Research* 52, 977-987, 2004.



Multiple Scenario Approach

- Multiple Scenario Approach (MSA) is an extension of MPA.
- The distributions of the stochastic variables are sampled to produce planned routes that include both known and stochastic requests.
 - Leave room in the routes to accommodate future requests.
- The real plans are obtained by projection over known requests only.



Double Horizon

- Mitrović-Minić S., "The Dynamic Pick-up and Delivery Problem with Time Windows", Ph.D. Thesis, School of Computing Science, Simon Fraser University, Canada, 2001.
- Mitrović-Minić S., R. Krishnamurti, G. Laporte, "Double-Horizon Based Heuristics for the Dynamic Pickup and Delivery Problem with Time Windows", *Transportation Research B* 38, 669-685, 2004.



Double Horizon

- Both a short-term and a long-term horizon are considered.
- A different objective is associated with each horizon.
 - Short term: “true” objective.
 - Long term: aimed at introducing large waiting times in the routes to favor the insertion of future requests.



Waiting Strategies

- Wait at current customer location if waiting time would be incurred by moving to next customer.



Waiting Strategies

- Mitrović-Minić S., "The Dynamic Pick-up and Delivery Problem with Time Windows", Ph.D. Thesis, School of Computing Science, Simon Fraser University, Canada, 2001.
- Mitrović-Minić S., G. Laporte, "Waiting Strategies for the Dynamic Pickup and Delivery Problem with Time Windows", *Transportation Research B* 38, 635-655, 2004.



Waiting Strategies

- PDPTW with a variable fleet size.
- Drive-First (DF): quit current location at the earliest possible time.
- Wait-First (WF): wait at current location as long as it is feasible.
 - potential for shorter routes, given that more requests are known when the vehicles leave their current location (more optimization opportunities).



Waiting Strategies

- Dynamic waiting strategies that combine DF and WF provide the best results.
- Best strategy: dynamic partition of a route into segments made of close locations.
 - Within a segment: Drive First.
 - When a boundary between two segments is to be crossed: wait at current location for a fraction of the time available up to the latest possible departure time.



Waiting Strategies

- Branke J., M. Middendorf, G. Noeth, M. Dessouky, "Waiting Strategies for Dynamic Vehicle Routing", *Transportation Science* 39, 298-312, 2005.



Waiting Strategies

- VRP with no time windows but with a time deadline.
- Given a set of planned routes, how should the slack time be distributed along the routes to maximize the probability of being able to service a new request?
- In the case of multiple vehicles, the best empirical strategy is:
 - "drive without waiting until the time to drive the remaining distance to the depot is equal to the slack time. Then, distribute the slack time to the remaining customers in proportion to the remaining driving distances".



Waiting Strategies

- Larsen A., O.B.G. Madsen, M.M. Solomon, "The A Priori Dynamic Traveling Salesman Problem with Time Windows", *Transportation Science* 38, 459-472, 2004.
- Relocation
 - Rather than waiting in a passive way at current customer location, you can move to some appropriate idle point in anticipation of future requests.



Waiting Strategies

- Ichoua S., M. Gendreau, J.-Y. Potvin,
"Exploiting Knowledge about Future
Demands for Real-Time Vehicle Dispatching",
forthcoming in *Transportation Science*.



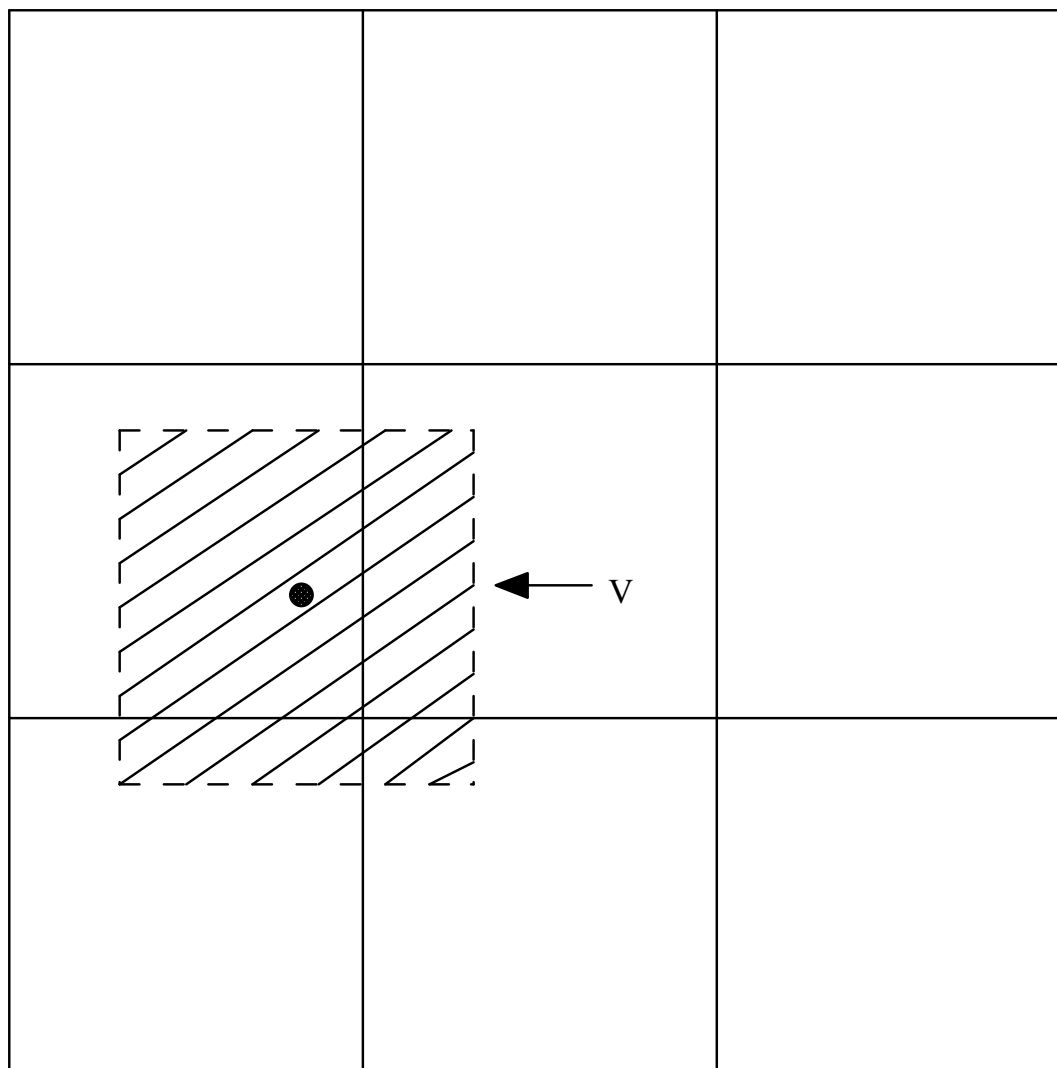
Assumptions

- Geographical area is divided into J zones, Z_1, Z_2, \dots, Z_J .
- Time horizon is partitioned into K time periods T_1, T_2, \dots, T_K .
- Request arrivals follow a Poisson process with rate λ_{jk} for zone Z_j and time period T_k .



Waiting rule

- d^{avg} is the average distance between two customer locations in the planned routes.
- d is the distance between current vehicle location and next customer location.
- $p_{\delta t, V}$ is the probability that at least one request will occur in the neighborhood V of current vehicle location in the next δt time units.
- n is the number of vehicles in the zone.





Waiting rule

IF

$(d > \alpha d^{\text{avg}})$ and $(p_{\delta t, V} > P)$ and $(n \leq N)$

THEN

wait for δt time units

OTHERWISE

go to next destination.



Waiting time

- Staying idle may be harmful, because it induces service delays in the planned route.
- Waiting time δt should be appropriately set.



Waiting time

$$\delta t = \min (t_1 - t, t_2 - t + \eta, t_3)$$

where

- t current time
- t_1 latest feasible departure time
- t_2 latest departure time without additional lateness
- t_3 threshold
- η tolerance to lateness



Waiting rule

- Impact the way events are handled.
- End of service
 - the rule is applied to determine if the vehicle should wait or not;
 - if it waits, a dummy customer is created at the current location;
 - the dummy is removed at time $t+\delta t$ if a true customer request does not occur.



Waiting rule

- New customer request
 - consider all dummies in the neighborhood and select the closest one;



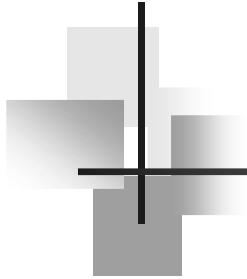
Computational results

- Simulation of 5 hours with 3 time periods (2 hours, 1 hour, 2 hours).
- 3X3 grid (for a total of 9 zones).
- Intense central zone.
- Two scenarios with (approximately) 125 and 175 dynamic requests.
- Fleet of 4 and 6 vehicles.



Computational results

- Results show the benefits obtained by anticipating demands:
 - Improvement of 4% on average with regard to total travel time and lateness.



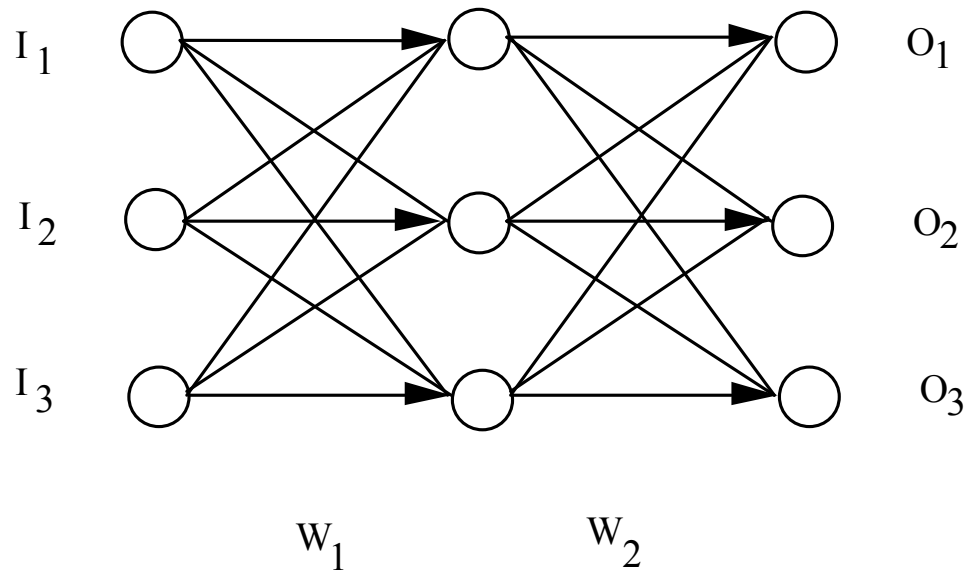
Neural Network for Vehicle Dispatching



Neural networks

- Can learn to perform a task from decision examples (provided by an expert).
- No need to devise an objective function.
- Human experts must perform well.

Neural networks





Neural networks

- Training set of examples (supervised learning).
 - (input, desired output) pair
- The neural network adjusts its connection weights to best approximate the required mapping between inputs and outputs.
- Each example is provided in turn to the neural network, which uses the desired output for incrementally adjusting its weights.
- Once training is completed, the neural network is tested against a new set of examples.



Neural networks

- A training example corresponds to a dispatching situation:
 - a planned route for each vehicle
 - a new request



Neural networks

- Input:
 - Vector of attributes that characterize the planned route of a vehicle with regard to the new request:
 - detour to service the new request
 - pick-up and delivery time of the new request
 - additional lateness...
- Desired output:
 - 1 if vehicle selected by human dispatcher
 - 0 otherwise
- Output (at the end of training)
 - « evaluation » of vehicle by neural network



Neural networks

- Once trained, the neural network is tested on a new set of examples.
- For each testing example (request):
 - rank vehicles from the one with largest evaluation (rank 1) to the one with smallest evaluation.
 - look at rank of vehicle selected by human dispatcher



Neural networks

Rank	Number of Decisions
1	20
2	14
3	6
4	7
5	1
6	0
7	1
8	1
9	0
10	0
11	0
12	0



Conclusion

- Dynamic travel times (and other types of events)
- Exploiting knowledge about the future
- Arc routing problems