Outline

1. Genetic and memetic algorithms
2. Application to the Vehicle Routing Problem (VRP)
3. MA|PM and the Heterogeneous Fleet VRP (HFVRP)
4. The Periodic VRP (PVRP).
PART 1

Genetic and Memetic Algorithms (GA and MA)
Classical GA – Incremental version

1: initialize population $Pop$
2: repeat
3: select two parents $P_1, P_2$ from $Pop$
4: crossover $P_1 \otimes P_2 \rightarrow C_1, C_2$
5: for each child $C$ do
6: mutate $C$ (small probability)
7: replace one solution $B$ in $Pop$ by $C$
8: endfor
9: until stopping criterion satisfied.

Not aggressive enough for combinatorial optimization, Cannot compete with tabu search for instance.
A simple example: the TSP 1/3

Travelling Salesman Problem
Data: $n$ nodes, distance matrix $D_{n \times n}$
Goal: compute one cycle of minimum total length, visiting each node once.

1. Chromosome and evaluation

One solution can be coded as a node permutation $C$. Example: $C = (5, 1, 4, 6, 3, 2)$
The fitness (solution cost) is:

$$F(C) = \sum_{i=1}^{n-1} D(C_i, C_{i+1}) + D(C_n, C_1)$$
A simple example: the TSP  2/3

2. Initial population $Pop$ of $nc$ chromosomes
Generate $nc$ random permutations, compute their costs.

3. Selection of two parents $P_1, P_2$ in $Pop$
Binary tournament: randomly draw two chromosomes, keep the best as $P_1$. Repeat to get $P_2$.

4. Simple one-point crossover

| Parent 1:  | 1 3 9 2|7 4 5 8 6 |
| Parent 2:  | 8 3 4 7|6 1 9 5 2 |
| Child 1:   | 1 3 9 2|8 4 7 6 5 |
| Child 2:   | 8 3 4 7|1 9 2 5 6 |
A simple example: the TSP 3/3

5. Simple mutation: swap two nodes
   \[
   \text{Child} : 1 \ 3 \ 9 \ 2 \ 7 \ 4 \ 1 \ 8 \ 6 \\
   \text{Mutated:} \ 1 \ 7 \ 9 \ 2 \ 3 \ 4 \ 1 \ 8 \ 6
   \]

6. Replacement rule
   Child C replaces the worst chromosome in Pop.

7. Stopping criteria
   - max number of crossovers
   - max number of crossovers without improvement etc.
Memetic Algorithm (Moscato, 1989)

Also called: Hybrid GA, Genetic Local Search.
Very effective. Requires a local search procedure $LS$.

1: \textit{initialize} population $Pop$
2: \textit{improve} each $s$ in $Pop$: $s \leftarrow LS(s)$
3: \text{repeat}
4: \text{select} two parents $P_1$, $P_2$ from $Pop$
5: \text{crossover} $P_1 \otimes P_2 \rightarrow C_1$, $C_2$
6: \text{for each} child $C$ \text{do}
7: \text{improve:} $C \leftarrow LS(C)$
8: \text{mutate} $C$ (small probability)
9: \text{replace} one solution $B$ in $Pop$ by $C$
10: \text{endfor}
11: \text{until} stopping criterion satisfied
PART 2

The VRP case
The Vehicle Routing Problem

Data:
- network with $n + 1$ nodes
- node 0: depot with identical vehicles of capacity $W$
- nodes 1 to $n$: clients with known demands $q_i$
- matrix $D$ of minimal traveling times (shortest paths)
- service costs and maximum trip duration (optional)

Goal: compute a set of trips of minimum total cost.

Notes:
- no split delivery
- the number of trips or vehicles is a decision variable
- NP-hard (TSP if total demand $\leq W$)
The Vehicle Routing Problem

A Solution for the VRPNC6 dataset

Problem 6 of Christofides, 50 clients.
Entry points for VRP literature


Starting point

In the 90's: efficient GAs for the TSP and the VRP with time windows, but none for the VRP.

Gendreau et al. (1998): "Published GAs for the VRP cannot compete with the best tabu search methods."

Chromosomes with trip delimiters:

```
1 5 4 2 7 3 6 8
```

Repair procedures in case of vehicle capacity violations. Perturbing the genetic transmission of good patterns.

**Key-idea:** use chromosomes **without** trip delimiters.
Chromosome and crossover

A chromosome is a permutation of the $n$ clients. Implicit shortest paths between adjacent clients. **No trip delimiters**: giant tour for a vehicle with $W=\infty$.

Two TSP crossovers that can be reused (we took OX):

- **LOX** random cut-points
- **OX** random cut-points

<table>
<thead>
<tr>
<th>P1</th>
<th>1 3 2</th>
<th>6 4 5</th>
<th>9 7 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>P2</td>
<td>3 7 8</td>
<td>1 4 9</td>
<td>2 5 6</td>
</tr>
<tr>
<td>C</td>
<td>3 7 8</td>
<td>6 4 5</td>
<td>1 9 2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P1</th>
<th>1 3 2</th>
<th>6 4 5</th>
<th>9 7 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>P2</td>
<td>3 7 8</td>
<td>1 4 9</td>
<td>2 5 6</td>
</tr>
<tr>
<td>C</td>
<td>8 1 9</td>
<td>6 4 5</td>
<td>2 3 7</td>
</tr>
</tbody>
</table>

But how to get a **VRP** solution and its cost?
Evaluation: procedure SPLIT

Chromosome $S = (a, b, c, d, e)$

Optimal splitting, cost 205

Auxiliary graph of possible trips for $W=10$ and shortest path in boldface
(Bellman's algorithm for directed acyclic graphs)
Comments 1/2

Practical advantages:
- repair procedures with trip delimiters are avoided
- classical crossovers for the TSP can be reused

No loss of information:
- there exists one optimal chromosome (permutation)
- the MA explores a smaller space (permutations)
- SPLIT evaluates each permutation optimally

Small price to pay:
- SPLIT runs in $O(n^2)$.
Flexibility of SPLIT:

- Constraints on trips do not affect the shortest path computation (e.g., maximum working time per driver).
- Hierarchic optimization of total cost ($F_1$) and number of trips ($F_2$): $\min M.F_1 + F_2$. The contrary is also possible.
- Limited fleet of $K$ vehicles. The general Bellman's algorithm gives at iteration $k$ the shortest paths with at most $k$ arcs. Use it and stop at iteration $K$.
- Vehicle Fleet Mix Problem or several vehicle types with different capacities and fixed costs: add to each arc cost the cost of the cheapest compatible vehicle type.
Population structure

Table $Pop$ of $nc$ distinct chromosomes:

- to avoid premature convergence ($clones$)
- better $dispersal$ of solutions (better $exploration$)

A simple cost-spacing rule for a stronger dispersal:

- $\forall P_1, P_2 \in Pop : |cost(P_1) - cost(P_2)| \geq \Delta$
- a child violating this rule is $rejected$
- acceptable rejection rate if $nc$ small, e.g. $nc = 30$

Initial composition of $Pop$:

- 3 good solutions obtained by classical VRP heuristics
- $nc$-3 random permutations, evaluated by SPLIT
Local search  1/2

Or-OPT (move 1 or 2 nodes)

2-OPT (change 2 edges)

Node exchanges are also considered.
Local search 2/2

Moves are also applied to 2 trips, e.g. 2-OPT.

All these moves can be scanned in $O(n^2)$.

First improvement instead of best improvement.
MA overview for the VRP

1: build Pop to get nc cost-spaced chromosomes
2: for phase := 1 to phase_max do
3:   sort Pop in ascending cost order (best soln Pop(1))
4:   repeat
5:     select $P_1, P_2$ by binary tournament
6:     apply crossover OX to get one child $C$
7:     if random < $P_{LS}$ then $C \leftarrow LS(C)$ endif
8:     select $B$ at random in the worst half of Pop
9:     if Pop\{$B$} $\cup$ $C$ is cost-spaced then $B \leftarrow C$ endif
10: until stopping criterion
11: partial_renewal (Pop)
12: endfor
## MA vs. classical VRP metaheuristics

14 Christofides instances, \( n=50-199 \)
Results obtained with various settings of parameters

<table>
<thead>
<tr>
<th>Kind</th>
<th>Authors</th>
<th>Year</th>
<th>Dev. to best solns</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS</td>
<td>Taillard</td>
<td>1993</td>
<td>0.05 %</td>
</tr>
<tr>
<td>MA</td>
<td>Prins</td>
<td>2001</td>
<td>0.08 %</td>
</tr>
<tr>
<td>TS</td>
<td>Gendreau et al.</td>
<td>1994</td>
<td>0.20 %</td>
</tr>
<tr>
<td>TS</td>
<td>Taillard</td>
<td>1992</td>
<td>0.39 %</td>
</tr>
<tr>
<td>TS</td>
<td>Rego and Roucairol</td>
<td>1996</td>
<td>0.55 %</td>
</tr>
<tr>
<td>TS</td>
<td>Gendreau et al.</td>
<td>1991</td>
<td>0.68 %</td>
</tr>
<tr>
<td>TS</td>
<td>Rego and Roucairol</td>
<td>1996</td>
<td>0.77 %</td>
</tr>
<tr>
<td>TS</td>
<td>Gendreau et al.</td>
<td>1994</td>
<td>0.86 %</td>
</tr>
<tr>
<td>TS</td>
<td>Osman</td>
<td>1993</td>
<td>1.01 %</td>
</tr>
<tr>
<td>SA</td>
<td>Osman</td>
<td>1993</td>
<td>2.09 %</td>
</tr>
</tbody>
</table>

Current best VRP metaheuristics 1/2

14 Christofides instances, \( n=50-199 \)
Results obtained with one setting of parameters
Times in minutes scaled for a 1 GHz PC

<table>
<thead>
<tr>
<th>Kind</th>
<th>Authors</th>
<th>Year</th>
<th>Gap%</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLS+ES</td>
<td>Mester, Bräysy</td>
<td>2004</td>
<td>0.03</td>
<td>7.72</td>
</tr>
<tr>
<td>GLS+ES fast</td>
<td>Mester, Bräysy</td>
<td>2004</td>
<td>0.07</td>
<td>0.27</td>
</tr>
<tr>
<td>Adaptive memory</td>
<td>Tarantilis, Kiranoudis</td>
<td>2002</td>
<td>0.23</td>
<td>1.04</td>
</tr>
<tr>
<td>MA</td>
<td>Prins</td>
<td>2001</td>
<td>0.24</td>
<td>2.60</td>
</tr>
<tr>
<td>RTR</td>
<td>Li et al.</td>
<td>2004</td>
<td>0.41</td>
<td>?</td>
</tr>
<tr>
<td>MA</td>
<td>Berger, Barkaoui</td>
<td>2004</td>
<td>0.49</td>
<td>4.25</td>
</tr>
<tr>
<td>TS</td>
<td>Cordeau et al.</td>
<td>2001</td>
<td>0.56</td>
<td>24.62</td>
</tr>
<tr>
<td>Granular TS</td>
<td>Toth, Vigo</td>
<td>2003</td>
<td>0.64</td>
<td>0.38</td>
</tr>
</tbody>
</table>

Source: Cordeau et al., New heuristics for the VRP, 2005.
Current best VRP metaheuristics 2/2

20 large-scale instances from Golden et al., \( n=200-483 \)
Results obtained with one setting of parameters
Times in minutes scaled for a 1 GHz PC

<table>
<thead>
<tr>
<th>Kind</th>
<th>Authors</th>
<th>Year</th>
<th>Gap%</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLS+ES</td>
<td>Mester, Bräysy</td>
<td>2004</td>
<td>0.00</td>
<td>72.9</td>
</tr>
<tr>
<td>GLS+ES fast</td>
<td>Mester, Bräysy</td>
<td>2004</td>
<td>0.93</td>
<td>0.6</td>
</tr>
<tr>
<td>Adaptive memory</td>
<td>Tarantilis, Kiranoudis</td>
<td>2002</td>
<td>0.74</td>
<td>8.4</td>
</tr>
<tr>
<td>MA</td>
<td>Prins</td>
<td>2001</td>
<td>0.91</td>
<td>33.4</td>
</tr>
<tr>
<td>RTR</td>
<td>Li et al.</td>
<td>2004</td>
<td>1.05</td>
<td>?</td>
</tr>
<tr>
<td>TS</td>
<td>Cordeau et al.</td>
<td>2001</td>
<td>1.45</td>
<td>56.1</td>
</tr>
<tr>
<td>Granular TS</td>
<td>Toth, Vigo</td>
<td>2003</td>
<td>2.87</td>
<td>1.8</td>
</tr>
</tbody>
</table>

Reference

PART 3

Memetic Algorithms with Population Management
The HVRP case
MA|PM (Sörensen, 2003)

General principles
- small population of high-quality solutions
- local improvement
- diversity controlled via population management

Population management
- distance $d(x,y)$ between solutions, in solution space
- distance to a population $P$: $d_P(s) = \min \{d(s,x): x \in P\}$
- a new solution $s$ is added to $P$ if $d_P(s) \geq \Delta$
- the diversity parameter $\Delta$ can be dynamically adjusted
MA|PM model for routing problems

1: \textit{initialize} population $P$ and diversity parameter $\Delta$
2: \textbf{repeat}
3: \textit{select} parents $P_1, P_2$ from $P$
4: apply crossover OX to get one child $C$
5: if random < $P_{LS}$ then $C \leftarrow LS(C)$ endif
6: select $B$ at random in the worst half of $P$
7: \textbf{while} $d_P(C) < \Delta$ \textbf{do} \textbf{or} \textbf{if} $d_P(C) \geq \Delta$ \textbf{then}
8: \textit{mutate} $C$
9: \textbf{end while}
10: \textit{replace} a solution $B$ by $C$ \textbf{if} \textbf{else} discard $C$
11: update diversity parameter $\Delta$
12: \textbf{until} stopping criterion satisfied
Some advantages of MA|PM

Bridge the gap between MA and Scatter Search (SS).
Can be viewed as a "light" form of SS.
Active control of diversity thru population management.
MA-like structure: easier to implement than SS.
Easy upgrade of an existing MA.
The HVRP

VRP with heterogeneous fleet (HFVRP or HVRP):

- $t$ vehicle types
- type $k$: availability $a(k)$, capacity $Q(k)$
- fixed cost $f(k)$, cost per unit distance $v(k)$
- trip $T$ of length $L$ with vehicle $k$: $\text{cost}(T) = f(k) + v(k).L$

Goal: minimize total cost of trips, each trip being assigned to one compatible vehicle.

VRP: case $t = 1$
Vehicle Fleet Mix Problem (VFMP): all $a(k) = \infty$ (or = $n$)
True HVRP: limited $a(k)$.
SPLIT for the VRP and VFMP

Consider one chromosome and a possible trip $T$ of length $L$.

Classical VRP, splitting graph with $O(m)$ arcs (one per trip):

- $\text{cost}(T) = L$
- splitting in $O(m)$ using Bellman algorithm.

VFMP:

- $\text{cost}(T) = f(k) + v(k).L$, $k$ cheapest compatible vehicle type.
- The shortest path computation does not change, $O(m)$.
Example for the VFMP

Fleet with $t = 3$ vehicle types:

Trip load : 25
Trip length: 300

<table>
<thead>
<tr>
<th>Type $k$</th>
<th>$Q_k$</th>
<th>$F_k$</th>
<th>$V_k$</th>
<th>Assignment of $k$ to $(i,j)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>100</td>
<td>10</td>
<td>Capacity violated</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>200</td>
<td>20</td>
<td>$C_{ij} = 200 + 20 \times 300 = 6200$</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>250</td>
<td>30</td>
<td>$C_{ij} = 250 + 30 \times 300 = 9250$</td>
</tr>
</tbody>
</table>
SPLIT for the HVRP  1/3

True HVRP (limited $a(k)$):

- the cost of a trip depends on the vehicle assigned to it
- this can be modelled by parallel arcs

\[
\text{Arc cost } 6200 \text{ (vehicle type 2)} \\
\begin{array}{c}
i \\
\end{array} \\
\begin{array}{c}
j \\
\end{array} \\
\text{Arc cost } 9250 \text{ (vehicle type 3)}
\]

- feasible path: no more than $a(k)$ vehicles of type $k$
- shortest path problem with resource constraints!
SPLIT for the HVRP  2/3

The splitting problem can be infeasible:

\[ t=2, \ a_1=1 \ et \ Q_1=10, \ a_2=1 \ et \ Q_2=14 \]

\[
\begin{align*}
S & : 1 \ 2 \ 3 \ 4 \quad \rightarrow \text{feasible} \\
q & : 7 \ 2 \ 9 \ 4 \\
S' & : 1 \ 4 \ 2 \ 3 \quad \rightarrow \text{infeasible} \\
q & : 7 \ 4 \ 2 \ 9
\end{align*}
\]

Resource-constrained shortest path problem!
Dynamic programming method

Let \( F(j, x_1, x_2, ..., x_t) \) be:

- the cost of an optimal splitting of \((S_1, S_2, ..., S_j)\)
- with \( 0 \leq x_k \leq a_k \) vehicles of each type \( k \).

Our goal is to compute \( F(n, a_1, a_2, ..., a_t) \) and we have:

- \( F(0, x_1, x_2, ..., x_t) = 0 \), for each type \( k \) and all \( 0 \leq x_k \leq a_k \)
- \( \forall j > 0, F(j, x_1, x_2, ..., x_t) = \min \{ F(i, x_1, x_2, ..., x_{k-1}, ..., x_t) : i \leq j, \text{load}(i,j) \leq Q_k, x_k > 0 \} \)

Can be implemented in \( O(m.t.n^t) \) if \( t \leq n \).

Pseudo-polynomial. Polynomial for a fixed \( t \).
Distance measures 1/3

$X, Y$ two chromosomes with $n$ tasks (required nodes). No trip delimiters: distances for permutations.

**Hamming distance:**

- $D_H(X, Y) = \text{number of indices with } \neq \text{ values in } X, Y.$
  
  $D_H(X, Y) = \sum_{i=1}^{n} (X(i) \neq Y(i))$

- $X = (1,2,3,4,5)$ \hspace{1cm} $D_H(X, Y) = 3$
  
  $Y = (1,3,4,2,5)$

- Values in $[0, n]$. Can be computed in $O(n)$.

- Drawback: $D_H(X, Y) = n$ if $Y$ is a circular shift of $X$. 
"Broken pairs" distance:

- \( D_R(X,Y) = \text{nb of pairs } \{X(i), X(i+1)\} \text{ not adjacent in } Y. \)
- \( X = (1,2,3,4,5) \) \( \quad \) (2,3) and (4,5) are broken
- \( Y = (3,4,1,2,5) \) \( D_R(X,Y) = 2 \)
- Values in \([0,n-1]\). Can be computed in \(O(n)\).

This distance is the best for vehicle routing problems ... if our encoding without trip delimiters is used!
Distances measures 3/3

Levenshtein (or edit) distance $D_L$:

- $X$ and $Y$ viewed as strings of symbols
- $D_L(X,Y)$ minimum nb of operations to change $X$ into $Y$.
- Allowed operations; delete, insert, replace one symbol.
- Values in $[0,n]$.
- Can be computed in $O(n^2)$ using a DP method.

Example. $D_L("fruits","baits") = 3$:

fruits $\rightarrow$ bruits $\rightarrow$ buits $\rightarrow$ baits
Diversity control policy

In MA|PM, child $C$ is accepted in $P$ if $d_P(C) \geq \Delta$. Policies:

- $\Delta = $ constant, e.g. $\Delta = 1 \iff$ no clones
- growth from 1 to a maximum value
- growth with reset when best-known solution improved.
Results for VFMP  1/2

12 instances from Golden, \( n=20-100 \)
Fixed costs but no variable costs
Best-known solutions (BKS) as listed in Choi and Tcha (2007)
Best of 5 runs, times scaled for a 2.4 GHz PC

<table>
<thead>
<tr>
<th>Authors</th>
<th>Year</th>
<th>Method</th>
<th>Dev BKS</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choi, Tcha</td>
<td>2007</td>
<td>CG+BB</td>
<td>0.004%</td>
<td>150</td>
</tr>
<tr>
<td>Prins</td>
<td>To appear</td>
<td>MA</td>
<td>PM</td>
<td>0.04%</td>
</tr>
<tr>
<td>Taillard</td>
<td>1999</td>
<td>HCG</td>
<td>0.14%</td>
<td>32</td>
</tr>
<tr>
<td>Prins</td>
<td>To appear</td>
<td>MA</td>
<td>0.15%</td>
<td>12</td>
</tr>
<tr>
<td>Wassan, Osman</td>
<td>2002</td>
<td>TS</td>
<td>0.23%</td>
<td>18</td>
</tr>
<tr>
<td>Gendreau et al.</td>
<td>1999</td>
<td>TS</td>
<td>0.24%</td>
<td>18</td>
</tr>
</tbody>
</table>

Note: the MA|PM improves one best-known solution.
Results for VFMP  2/2

12 instances from Golden, \( n=20-100 \)
Fixed costs and variable costs
Best-known solutions (BKS) as listed in Choi and Tcha (2007)
Best of 5 runs, times scaled for a 2.4 GHz PC

<table>
<thead>
<tr>
<th>Authors</th>
<th>Year</th>
<th>Method</th>
<th>Dev BKS</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prins</td>
<td>To appear</td>
<td>MA</td>
<td>PM</td>
<td>-0.06%</td>
</tr>
<tr>
<td>Choi,Tcha</td>
<td>2007</td>
<td>CG+BB</td>
<td>0.00%</td>
<td>93</td>
</tr>
<tr>
<td>Prins</td>
<td>To appear</td>
<td>MA</td>
<td>0.01%</td>
<td>11</td>
</tr>
</tbody>
</table>

Note: the MA|PM improves 2 best-known solutions.
Results for the true HVRP

8 larger Golden instances, \( n = 50-100 \)
Best-known solutions (BKS) as listed in Tarantilis
One single run, times scaled for a 2.4 GHz PC

<table>
<thead>
<tr>
<th>Authors</th>
<th>Year</th>
<th>Method</th>
<th>Dev BKS</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Li, Golden, Wasil</td>
<td>2007</td>
<td>RTR</td>
<td>-0.49%</td>
<td>286</td>
</tr>
<tr>
<td>Prins</td>
<td>To appear</td>
<td>MA</td>
<td>PM</td>
<td>-0.45%</td>
</tr>
<tr>
<td>Prins</td>
<td>To appear</td>
<td>MA</td>
<td>-0.04%</td>
<td>50</td>
</tr>
<tr>
<td>Tarantilis et al.</td>
<td>2004</td>
<td>TA</td>
<td>0.09%</td>
<td>86</td>
</tr>
<tr>
<td>Taillard</td>
<td>1999</td>
<td>HCG</td>
<td>0.40%</td>
<td>52</td>
</tr>
</tbody>
</table>

The MA improves 4 best-known solutions.
The RTR and MA|PM improve 6 best-known solutions.
References


PART 4

Periodic problems
The Periodic VRP - Basic version

PVRP data:
- undirected graph $G$, depot + $n$ required nodes (tasks)
- planning horizon $H$ of $np$ periods or "days"
- for each task $i$, frequency $f(i)$ (number of visits in $H$)
- demands $q(i)$, service times $s(i)$, travel times $C(i,j)$.

Goal (hierarchical bi-objective function):
- select $f(i)$ days per task $i$ and solve one VRP per day
- main objective: minimize fleet size
- secondary objective: total duration of trips over $H$

Applications:
- industrial waste collection, drinks dispensers,
- delivery of propane to glass houses etc.
Varying demands & service times

Published models: $q(i), s(i)$ do not depend on period $p$.

But demands often result from daily "productions"

- constant production : $\text{prod}(i,p) = \text{prod}(i)$
- global variation $\alpha(p)$ : $\text{prod}(i,p) = \alpha(p) \cdot \text{prod}(i)$
- general case : distinct $\text{prod}(i,p), p=1...np$

If last visit in period $q$, the amount in period $p$ is:

$$q(i, p) = \sum_{k=q+1}^{p} \text{prod}(i,k)$$

Service time: $s(i,p)$ linear in $q(i,p) + \text{fixed cost}$. 
Spacing constraints

Two types (convertible into each other):

- min & max time lag between two visits
- set of allowed day combinations \( \text{comb}(i) \)

We use day combinations:

- not too many for small horizons (1 week → 1 month)
- spacing is implicitly satisfied
- demands can be computed in advance for each day

Example:

- cyclic horizon, \( np=7 \), no work on week-end, \( f(i)=1,2,5 \)
- 8 combinations: \{1,2,3,4,5\},\{1,4\},\{2,5\},\{1\}...\{5\}
Chromosomes

Each task $i$ has a set of day combinations.

Chromosome $S$:

- $np$ sublists $S(1)...S(np)$: one VRP chromosome per day
- node $i$ occurs $f(i)$ times, using one day combination
- node $i$ occurs at most once in each sublist $S(k)$

Chromosome length:

$$L = \sum_{i=1}^{n} f(i)$$

No trip delimiter: $S$ kind of priority order for one vehicle doing the tasks day by day and one by one in each day.
### Extended LOX crossover (ELOX)

<table>
<thead>
<tr>
<th></th>
<th>$f(i)$</th>
<th>$comb(i)$</th>
<th></th>
<th>$f(i)$</th>
<th>$comb(i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>{1, 3, 4}, {2, 3, 4}</td>
<td>5</td>
<td>1</td>
<td>{1}, {2}</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>{1, 3}, {2, 4}</td>
<td>6</td>
<td>3</td>
<td>{2, 3, 4}, {1, 2, 4}, {1, 2, 3}</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>{1}, {4}</td>
<td>7</td>
<td>2</td>
<td>{2, 4}, {1, 3}</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>{1}</td>
<td>8</td>
<td>1</td>
<td>{2}, {4}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>P2</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>6</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>C1</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>8</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>6</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
Chromosome evaluation

Hierarchical bi-objective function $Z = M.nvu + tcost$, where:

- $nvu$, number of vehicles used (fleet size required)
- $tcost$, total duration of trips over horizon $H$

Optimal chromosome evaluation:

1. SPLIT in each day $p$, with arc costs = 1 in auxiliary graph $\rightarrow$ minimum nb of vehicles per day $nv(p)$.
2. Minimum fleet size $nvu = \text{maximum of the } nv(p)$.
3. SPLIT on each day $p$ using actual trip costs and at most $nvu$ arcs $\rightarrow$ minimum cost per day $cost(p)$
4. Return $Z = M.nvu + \text{sum of the } cost(p)$. 
Role of the local search

We have to kinds of decisions in the MA:

- tactical level: change the days for some tasks
- operational level: change the trips in one given day

The MA is NOT a two-phase method, it can modify a solution at the two levels.

But, to limit running time, the local search is applied to each day separately (operational level, VRP local search).

The crossover deals with the tactical level.

In general, this kind of separation is fruitful in MAs.
The method was developed in fact for the periodic capacitated arc routing problem (PCARP) but the algorithm is easily transposed to the PVRP:


Concluding remarks

Memetic algorithms are **effective tools** to solve vehicle routing problems.

The SPLIT procedure is **general and flexible**... as long as the shortest path problem is polynomial (VRP, VRPTW, PVRP) or pseudo-polynomial (HVRP)!

Software engineering point of view: the progressive upgrade GA → MA → MA|PM is (relatively) easy.

Remarks confirmed by other applications not described here (e.g. time windows, multi-objective problems).