LOCAL SEARCH METHODS
APPLICATIONS AND ENGINEERING

Lecture 14

Further Notions in Optimisation

Marco Chiarandini

Outline

Outline

Specifics of the Experiment

- Objectives
- Source of Variance and their Type (factors)
- Instances
- Experimental Design
- Experimental Unit and Time Limit
- Performance measure
- Approximate Computational Power required
- Model and Scheme of Analysis

Outline

Optimization Under Uncertainty

In many real life cases some problem parameters are unknown. Hence the optimization is under uncertainty.

- If the parameters are known only within certain bounds, one approach to tackling such problems is called robust optimization.

  In robust optimization the goal is to find a solution which is feasible for all such data and optimal in some sense.

- If probability distributions governing the data are known or can be estimated then it is possible to take advantage of this and use stochastic optimization.

  In stochastic optimization the goal is to find some policy that is feasible for all (or almost all) the possible data instances and maximizes the expectation of some function of the decisions and the random variables.

Solutions to Stochastic Problems

- *A priori* solutions
- On line solutions
- Mixed strategy solutions (two-stage)

Two-stage solutions consists of:

- A first stage where some action is taken
- A second stage (made by recourse decisions) that compensates for any bad effects that might have been experienced as a result of the first-stage decision and the random outcome of events.

The optimal policy from such a model is a single first-stage policy and a collection of recourse decisions (a decision rule) defining which second-stage action should be taken in response to each random outcome.

Solution approaches

- The expected value of the objective function can be computed mathematically.

  $$g(s) = E[f(\pi, s)] \qquad s \in \mathcal{S} and \pi \text{stochastic variables}$$

  Then no difference with a deterministic problem. Although the evaluation function may become computationally prohibitive.

- The expected value of the objective function can only be estimated via sampling and simulation

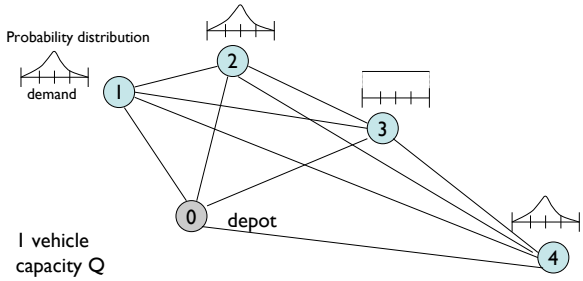
  $$g(s) = \overline{f(\pi, s)} = \frac{1}{N} \sum_{i=1}^{N} f(\pi_i, s) \qquad \text{(unbiased estimator)}$$

VRPSD introduction

- VRP: n customers, n demands, 1 depot, 1 vehicle, delivery of goods
- VRP objective: minimize total trevel cost
- The VRPSD: a VRP with *stochastic demands*
- The objective: minimize expected travel cost

An instance

exact demand is revealed only when the vehicle arrives at a customer



1 vehicle
capacity Q

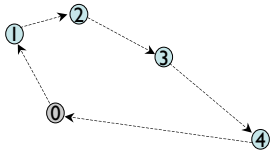
Solution strategies

1. a-priori, off-line
2. dynamic, on-line
3. mixed → we focus on the *restocking policy*

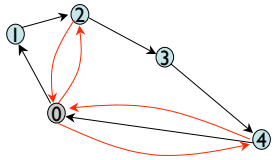
A-priori strategy

1. a-priori, off-line
2. dynamic, on-line
3. mixed

T, a-priori solution



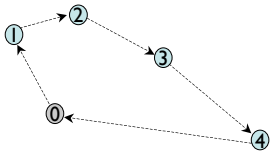
vehicle follows T, but
if not enough capacity → route failure



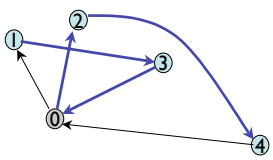
On-line strategy

1. a-priori, off-line
2. dynamic, on-line
3. mixed

T, a-priori solution



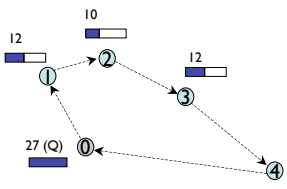
T may be completely modified by an algorithm which **re-optimizes** after each demand is known



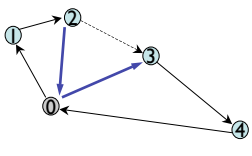
Mixed strategy: the restocking policy

1. a-priori, off-line
2. dynamic, on-line
3. mixed

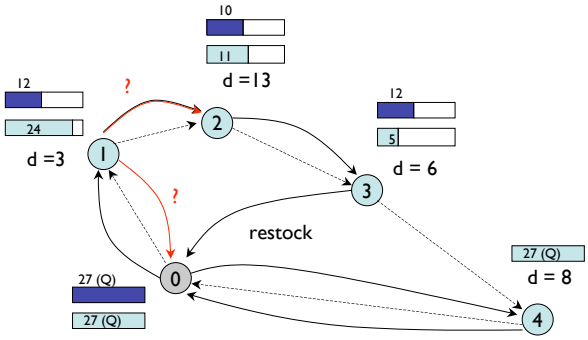
Pre-plannig: *tour* and *thresholds*



Restocking in strategic points



Restocking policy in action



Motivation for the "mixed strategy" approach

Secomandi (1998) shows that the gap between optimal on line and optimal mixed strategies is small (< 3%)

On line strategies are much more computationally demanding

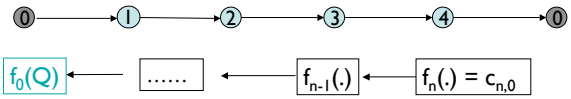
On line strategies require frequent re-planning of routes which may be not appealing for drivers

Objective function (1)

$f_j(q)$ = expected cost-to-go *from the customer T(j) on*, *after* having serviced T(j), given a *residual capacity* q

the objective function value is:

$f_0(Q)$ = expected cost of the a priori tour T

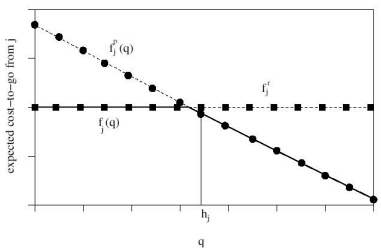


Objective function (2)

$$f_j(q) = \begin{cases} f_j^{proceed}(q), & \text{if } q \geq threshold_j \\ f_j^{restock}(q), & \text{otherwise} \end{cases}$$

Objective function (3)

the threshold:



Objective function (4)

- demands d_i, indipendent random variables
- p_{ik} = Prob($d_i = k$) , k = 0,1, …, K ≤ Q

$$f_j(q) = \min \left\{ f_j^{proceed}(q), \; f_j^{restock}(q) \right\}$$

$$f_j^{proceed}(q) = c_{j,j+1} + \sum_{k \leq q} f_{j+1}(q-k)\, p_{j+1,k} + \sum_{k>q} \left[2\, c_{j+1,0} + f_{j+1}(q-k) \right] p_{j+1,k}$$

$$F_j^{restock}(q) = c_{j,0} + c_{0,j+1} + \sum_{k} f_{j+1}(Q-k)\, p_{j+1,k}$$

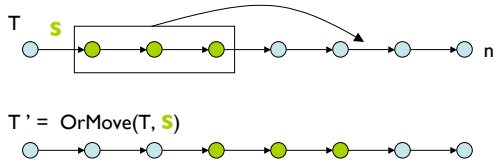
Time complexity of objective function evaluation

$O(nKQ)$

for (q = Q, Q-1, ..., 0) $f_n(q) = c_{n,0}$ // boundary condition of recursion
for (j = n-1, n-2, ..., 1) do
  compute $f_j^{restock}$ // using previously computed $f_{j+1}(.)$
  for (q = Q, Q-1, ..., 0) do
    compute $f_j^{proceed}(q)$ // using previously computed $f_{j+1}(.)$
    compare $f_j^{restock}$ and $f_j^{proceed}(q)$ for finding threshold h_j
    compute $f_j(q) = \min\{f_j^{proceed}(q), f_j^{restock}\}$
  end for
end for
compute $f_0(Q)$ // using previously computed $f_1(.)$
return $f_0(Q)$

The local search we adopted: Or-opt

the basic move:



the neighborhood of a sequence T: | S | = r, r = 3, 2, 1

why Or-opt: simple neighborhood allows fast exploration

Fast proxy delta function

Δ cost (T, OrMove(S))

=

Deletion Cost(S) + Insertion Cost(S)

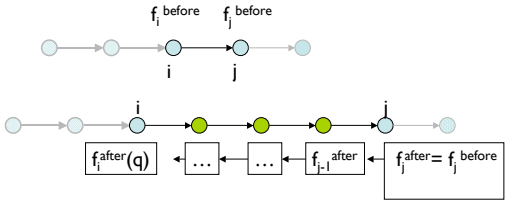
≅

Approximate Deletion Cost(S) + Approximate Insertion Cost(S)

Fast proxy delta function (2)

inserting *between i and j:*

Approximate Insertion Cost = $\sum_{q=0}^{Q} \frac{f_i^{after}(q) - f_i^{before}(q)}{Q+1}$ ⇒ O(KQ)



Time complexity

O(n) · O(complexity of Δ cost) → O(n^2KQ) exact Δ cost
→ O(nKQ) proxy Δ cost

T initial route
for (r =3, 2, 1) do{
 while (found better) {
  for each set S(r) {
   compute Δ cost (T, OrMove(T, S(r))) \\ try only 1 insertion point
  } **end for**
  choose S(r) leading to most negative Δ cost
  set T = OrMove(T, S(r))
 } **end while**
} **end for**