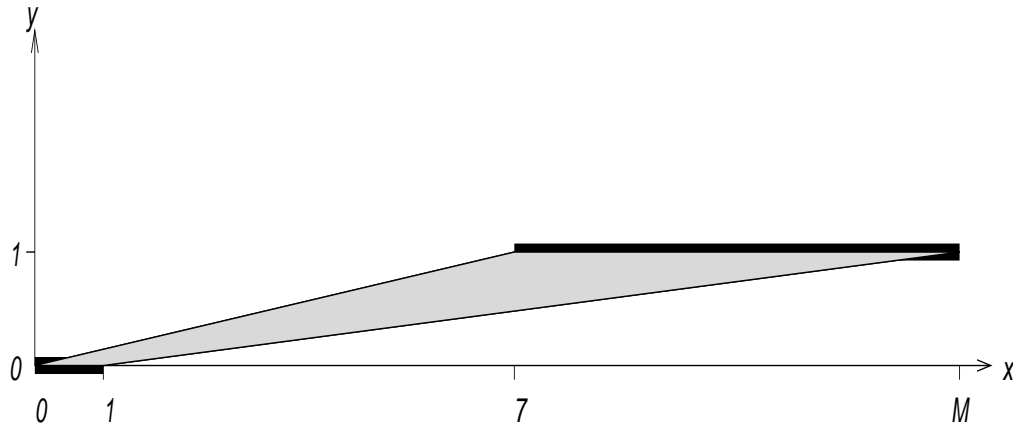


Lecture 11: Integer programming

When are integer models needed?

- Products or raw materials are indivisible
- Logical constraints: “if A then B ”; “ A or B ”
- Fixed costs
- Combinatorics (sequencing, allocation)
- On/off-decision to buy, invest, hire, generate electricity,
...

Either $0 \leq x \leq 1$ or $x \geq 7$



Let $M \gg 1$: $x \leq 1 + My$, $x \geq 7y$, $y \in \{0, 1\}$

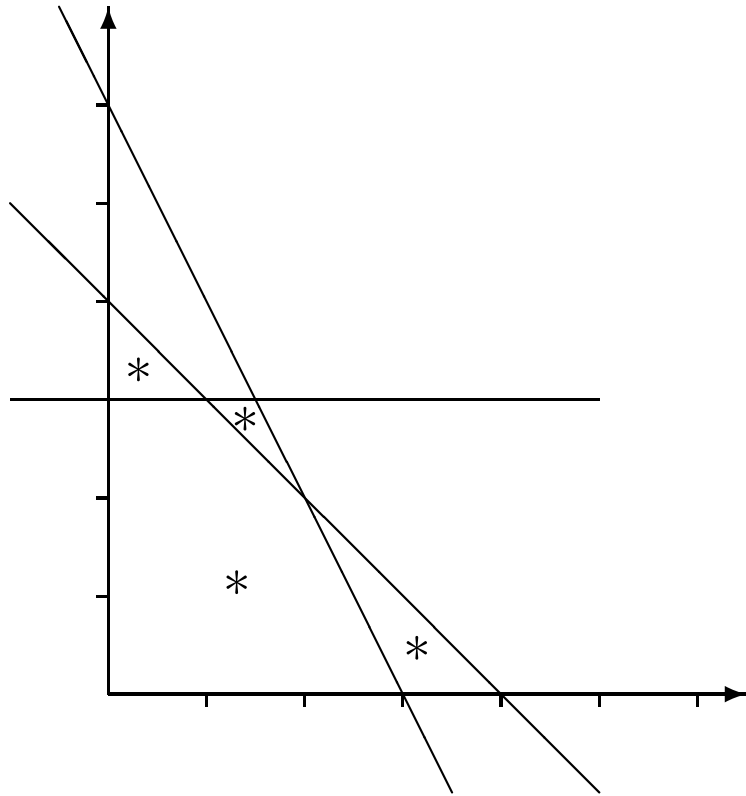
Variable x may only take the values 2, 45, 78 & 107

$$x = 2y_1 + 45y_2 + 78y_3 + 107y_4$$

$$y_1 + y_2 + y_3 + y_4 = 1$$

$$y_1, y_2, y_3, y_4 \in \{0, 1\}$$

At least 2 of 3 constraints must be fulfilled



* = feasible regions

$$M \geq 2$$

$$x_1 + x_2 \leq 4 \quad (1)$$

$$2x_1 + x_2 \leq 6 \quad (2)$$

$$x_2 \leq 3 \quad (3)$$

$$\text{and } x_1, x_2 \geq 0$$

$$x_1 + x_2 \leq 4 + M(1 - y_1) \quad (1)$$

$$2x_1 + x_2 \leq 6 + M(1 - y_2) \quad (2)$$

$$x_2 \leq 3 + M(1 - y_3) \quad (3)$$

$$y_1 + y_2 + y_3 \geq 2$$

$$y_1, y_2, y_3 \in \{0, 1\}$$

$$\text{and } x_1, x_2 \geq 0$$

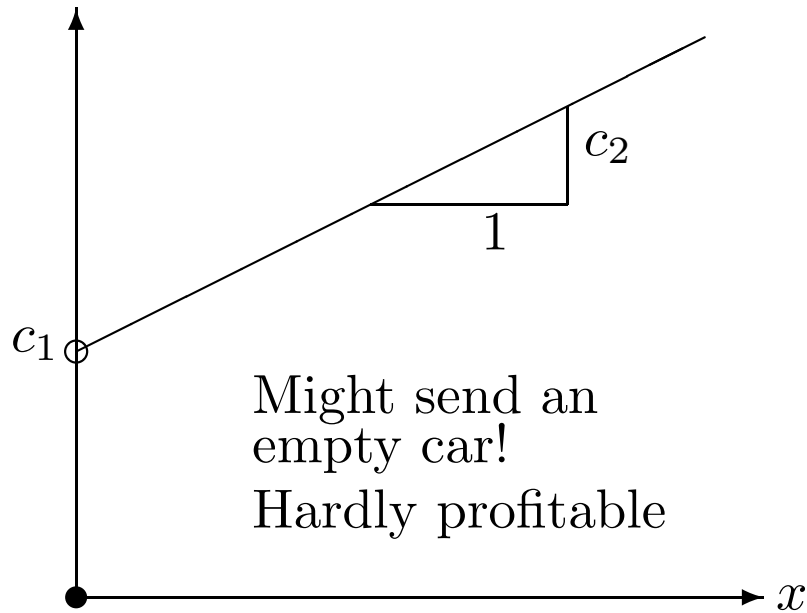
Fixed costs

x = the amount of a certain product to be sent

If $x > 0$ then the initial cost c_1 (e.g. car hire) is generated

Variable cost c_2 per unit sent

$$\text{Total cost: } f(x) = \begin{cases} 0 & \text{if } x = 0 \quad \boxed{\text{effect}} \\ c_1 + c_2 \cdot x & \text{if } x > 0 \quad \boxed{\text{wanted!}} \end{cases}$$



Let M = car capacity

$$y = \begin{cases} 1 & \text{if } x > 0 \quad \boxed{\text{effect}} \\ 0 & \text{if } x = 0 \quad \boxed{\text{wanted!}} \end{cases}$$

$$f(x, y) = c_1 \cdot y + c_2 \cdot x$$

$$x \leq M \cdot y \quad \boxed{\text{linear 0/1 model!}}$$

$$x \geq 0, \quad y \in \{0, 1\}$$

Other applications of integer optimization

- Facility location (new hospitals, shopping centers, etc.)
- Scheduling (on machines, personnel, projects, schools)
- Logistics (material- and warehouse control)
- Distribution (transportation of goods, buses for disabled persons)
- Production planning
- Telecommunication (network design, frequency allocation)
- VLSI design

The combinatorial explosion

Assign n persons to carry out n jobs # feasible solutions: $n!$

Assume that a feasible solution is evaluated in 10^{-9} seconds

n	2	5	8	10	100
$n!$	2	120	$4.0 \cdot 10^4$	$3.6 \cdot 10^6$	$9.3 \cdot 10^{157}$
[time]	10^{-8} s	10^{-6} s	10^{-4} s	10^{-2} s	10^{142} yrs

Complete enumeration of all solutions is **not** an efficient algorithm!

An algorithm exists that solves this problem in time $\mathcal{O}(n^4) \propto n^4$

n	2	5	8	10	100	1000
n^4	16	625	$4.1 \cdot 10^3$	10^4	10^8	10^{12}
[time]	10^{-7} s	10^{-6} s	10^{-5} s	10^{-5} s	10^{-1} s	17 min

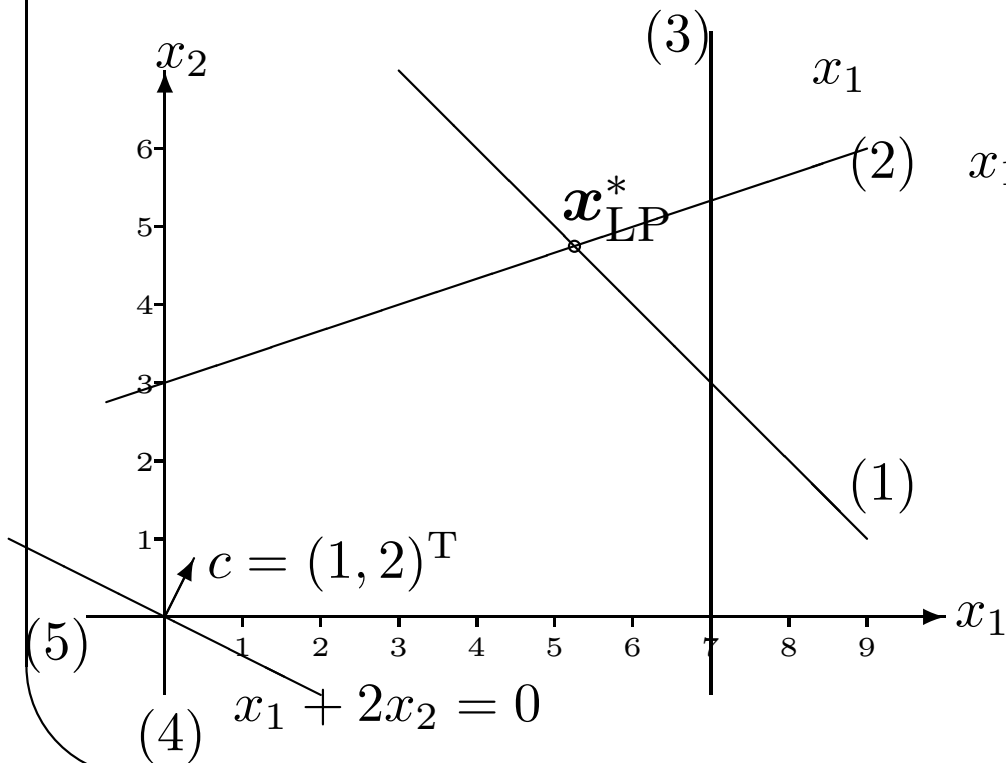
Linear continuous optimization model

$$\begin{aligned} \max \quad z_{\text{LP}} &= x_1 + 2x_2 \\ \text{s.t.} \quad & x_1 + x_2 \leq 10 \quad (1) \end{aligned}$$

$$-x_1 + 3x_2 \leq 9 \quad (2)$$

$$x_1 \leq 7 \quad (3)$$

$$x_1, x_2 \geq 0 \quad (4, 5)$$



$$x_{\text{LP}}^* = \begin{pmatrix} 21/4 \\ 19/4 \end{pmatrix}$$

$$z_{\text{LP}}^* = 14\frac{3}{4}$$

Linear integer optimization model

$$\max z_{\text{IP}} = x_1 + 2x_2$$

$$\text{s.t.} \quad x_1 + x_2 \leq 10 \quad (1)$$

$$-x_1 + 3x_2 \leq 9 \quad (2)$$

• = feasible
integer points

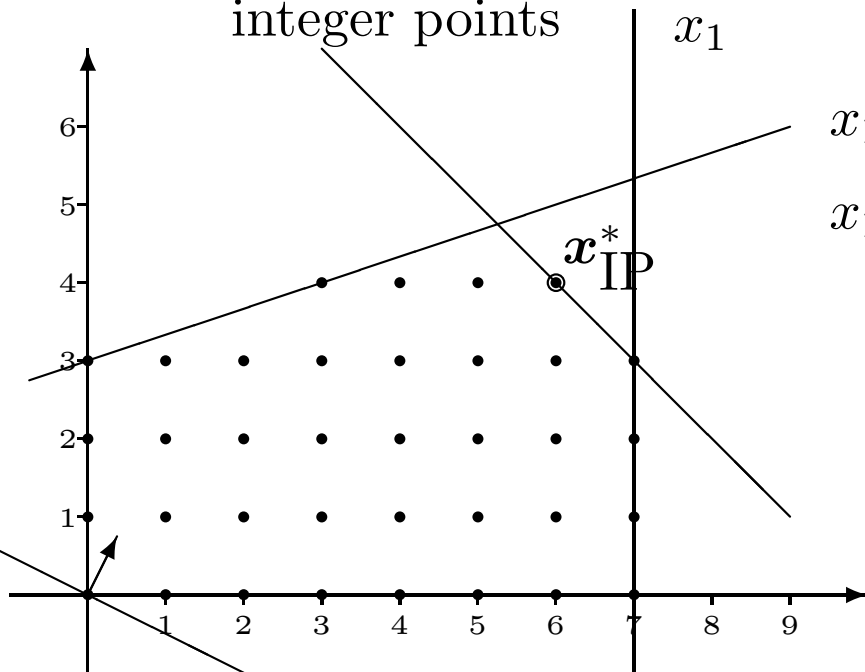
$$x_1 \leq 7 \quad (3)$$

$$x_1, x_2 \geq 0 \quad (4, 5)$$

x_1, x_2 integer

$$\mathbf{x}_{\text{IP}}^* = \begin{pmatrix} 6 \\ 4 \end{pmatrix}$$

$$z_{\text{IP}}^* = 14 < z_{\text{LP}}^*$$

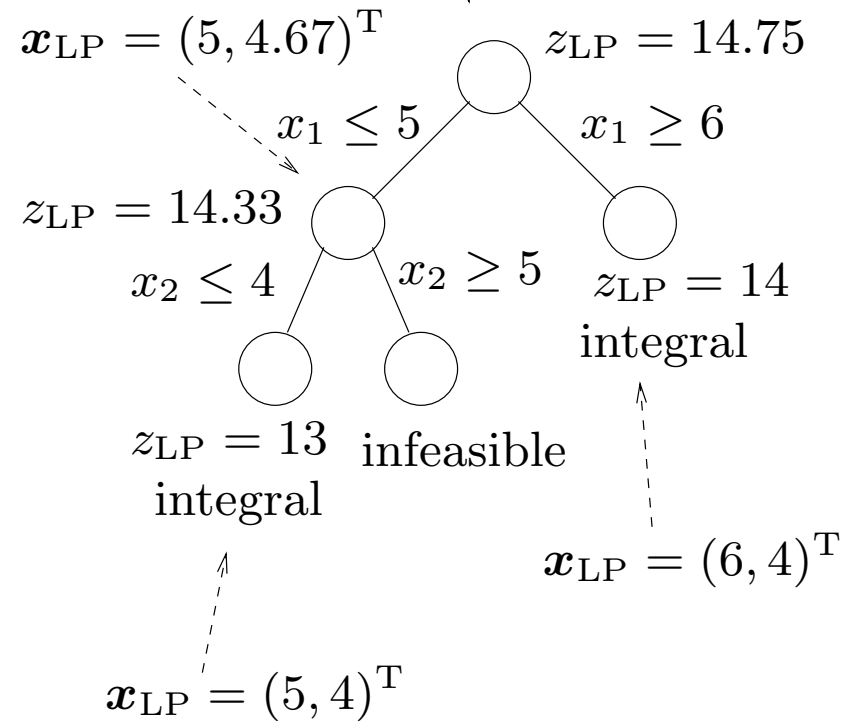
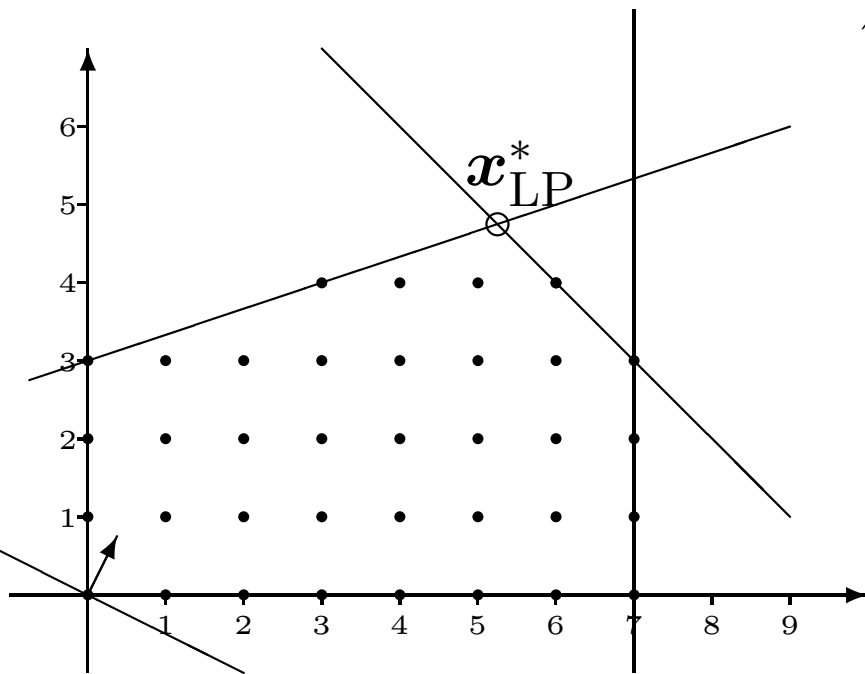


Classic methods

- *Branch-and-Bound*: relaxation plus divide-and-conquer
- *Cutting plane method*: relaxation plus generations of constraint that cut off infeasible (e.g., non-integer) points generated
- “Relaxation” can be the continuous or Lagrangian one
- *Lagrangian optimization*: Lagrangian relaxation plus multiplier optimization
- These methods are often combined (e.g., cutting planes added at nodes in B&B tree: Branch & Cut)

The branch-and-bound-algorithm

Relax integrality constraints \Rightarrow linear program $\Rightarrow \mathbf{x}_{LP}^* = (5.25, 4.75)^T$



Cutting plane methods

- Goal: generate the convex hull of the feasible integer vectors
- Result: Can solve the IP by solving the LP relaxation over this convex hull
- Compare IP example: one extra linear constraint defines the entire convex hull! ($x_2 \leq 4$)
- Means: Relax problem (e.g., continuous relaxation); Solve. If infeasible solution, generate constraint to the relaxation that cuts off that vector but no feasible vectors. Repeat
- Constraint generation called a *separation oracle*

The complexity of integer optimization, I

The Mexico LP has (in the version which is handed out) 113 variables and 84 linear constraints. Solution by a slow (333 MHz Unix) computer: 0.01 s

IP variant: add a fixed cost for using a railway link for the raw material transport \implies 78 0/1 variables

Fixed cost 100 \Rightarrow 20 s.

18,000 B & B nodes

60,000 simplex iterations

300 \Rightarrow 3 min.

208,000 B & B nodes

650,000 simplex iterations

Notes:

- * There are $2^{78} \approx 0.3 \cdot 10^{24}$ possible combinations. B & B is good at implicitly enumerating them all ...

* The higher the fixed cost, the more difficult the problem. Why?

Continuous relaxation worse approximation

The complexity of integer optimization, II

- A variation of the knapsack problem (and more difficult)
- Each variable has a value and “weight” per unit
- AMPL model:

```
var x{1..5} integer, >=0;
```

```
maximize ka:213*x[1]-1928*x[2]-11111*x[3]-2345*x[4]+9123*x[5];
```

```
subject to c1:
```

```
12223*x[1]+12224*x[2]+36674*x[3]+61119*x[4]+85569*x[5] =
```

```
89643482;
```

- Often binary; here, general integer variables
- The problem has an optimal solution

- LP relaxation trivial: sort variables in descending order of c_j/a_j ; take the best one; fill the knapsack exactly
- LP solution? $\mathbf{x}_{\text{LP}}^* \approx (0, 0, 0, 0, 1047.62)^T$
- Optimal solution: $\mathbf{x}^* = (7334, 0, 0, 0, 0)^T$
- IP solution? Report after roughly 3 hours of CPU time:

Mixed integer rounding cuts applied: 2

Gomory fractional cuts applied: 1

CPLEX 10.1.0: unrecoverable failure with no integer solution.

20298576 MIP simplex iterations

384198302 branch-and-bound nodes; no basis.

x [*] := 0 0 0 0 0 ;

The Philips example—TSP solved heuristically

- Let c_{ij} denote the distance between cities i and j , with

$\{i, j\} \subset \mathcal{N}$ — set of nodes

$(i, j) \in \mathcal{L}$ — set of links

- Links (i, j) and (j, i) the same; direction plays no role

- $x_{ij} = \begin{cases} 1, & \text{if link } (i, j) \text{ is part of the TSP tour,} \\ 0, & \text{otherwise} \end{cases}$

- The Traveling Salesman Problem (TSP):

$$\text{minimize} \quad \sum_{(i,j) \in \mathcal{L}} c_{ij} x_{ij}$$

$$\text{subject to} \quad \sum_{(i,j) \in \mathcal{L}: \{i,j\} \subset \mathcal{S}} x_{ij} \leq |\mathcal{S}| - 1, \quad \mathcal{S} \subset \mathcal{N}, \quad (1)$$

$$\sum_{(i,j) \in \mathcal{L}} x_{ij} = n, \quad (2)$$

$$\sum_{i \in \mathcal{N}: (i,j) \in \mathcal{L}} x_{ij} = 2, \quad j \in \mathcal{N}, \quad (3)$$

$$x_{ij} \in \{0, 1\}, \quad (i, j) \in \mathcal{L}$$

Interpretations

- Constraint (1) implies that there can be no *sub-tours*, that is, a tour where fewer than n cities are visited (that is, if $\mathcal{S} \subset \mathcal{N}$ then there can be at most $|\mathcal{S}| - 1$ links between nodes in the set \mathcal{S} , where $|\mathcal{S}|$ is the cardinality–number of members of–the set \mathcal{S});
- Constraint (2) implies that in total n cities must be visited;
- Constraint (3) implies that each city is connected to two others, such that we make sure to arrive from one city and leave for the next

Lagrangian relaxation

- TSP is NP-hard—no known polynomial algorithms exist
- Lagrangian relax (3) for all nodes except starting node
- Remaining problem: 1-MST—find the minimum spanning tree in the graph without the starting node and its connecting links; then, add the two cheapest links to connect the starting node
- Starting node $s \in \mathcal{N}$ and connected links assumed removed from the graph

- Objective function of the Lagrangian problem:

$$\begin{aligned}
 q(\boldsymbol{\lambda}) &= \underset{\boldsymbol{x}}{\text{minimum}} \sum_{(i,j) \in \mathcal{L}} c_{ij} x_{ij} + \sum_{j \in \mathcal{N}} \lambda_j \left(2 - \sum_{i \in \mathcal{N}: (i,j) \in \mathcal{L}} x_{ij} \right) \\
 &= 2 \sum_{j \in \mathcal{N}} \lambda_j + \underset{\boldsymbol{x}}{\text{minimum}} \sum_{(i,j) \in \mathcal{L}} (c_{ij} - \lambda_i - \lambda_j) x_{ij}
 \end{aligned}$$

- A high (low) value of the multiplier λ_j makes node j attractive (unattractive) in the 1-MST problem, and will therefore lead to more (less) links being attached to it
- Subgradient method for updating the multipliers

- Updating step:

$$\lambda_j := \lambda_j + \alpha \left(2 - \sum_{i \in \mathcal{N}: (i,j) \in \mathcal{L}} x_{ij} \right), \quad j \in \mathcal{N},$$

where $\alpha > 0$ is a step length

- Update means:

Current degree at node j :

$$\left\{ \begin{array}{l} > 2 \implies \lambda_j \downarrow \text{ (link cost } \uparrow \text{)} \\ = 2 \implies \lambda_j \leftrightarrow \text{ (link cost constant)} \\ < 2 \implies \lambda_j \uparrow \text{ (link cost } \downarrow \text{)} \end{array} \right.$$

- Link cost shifted upwards (downwards) if too many (too few) links connected to node j in the 1-MST

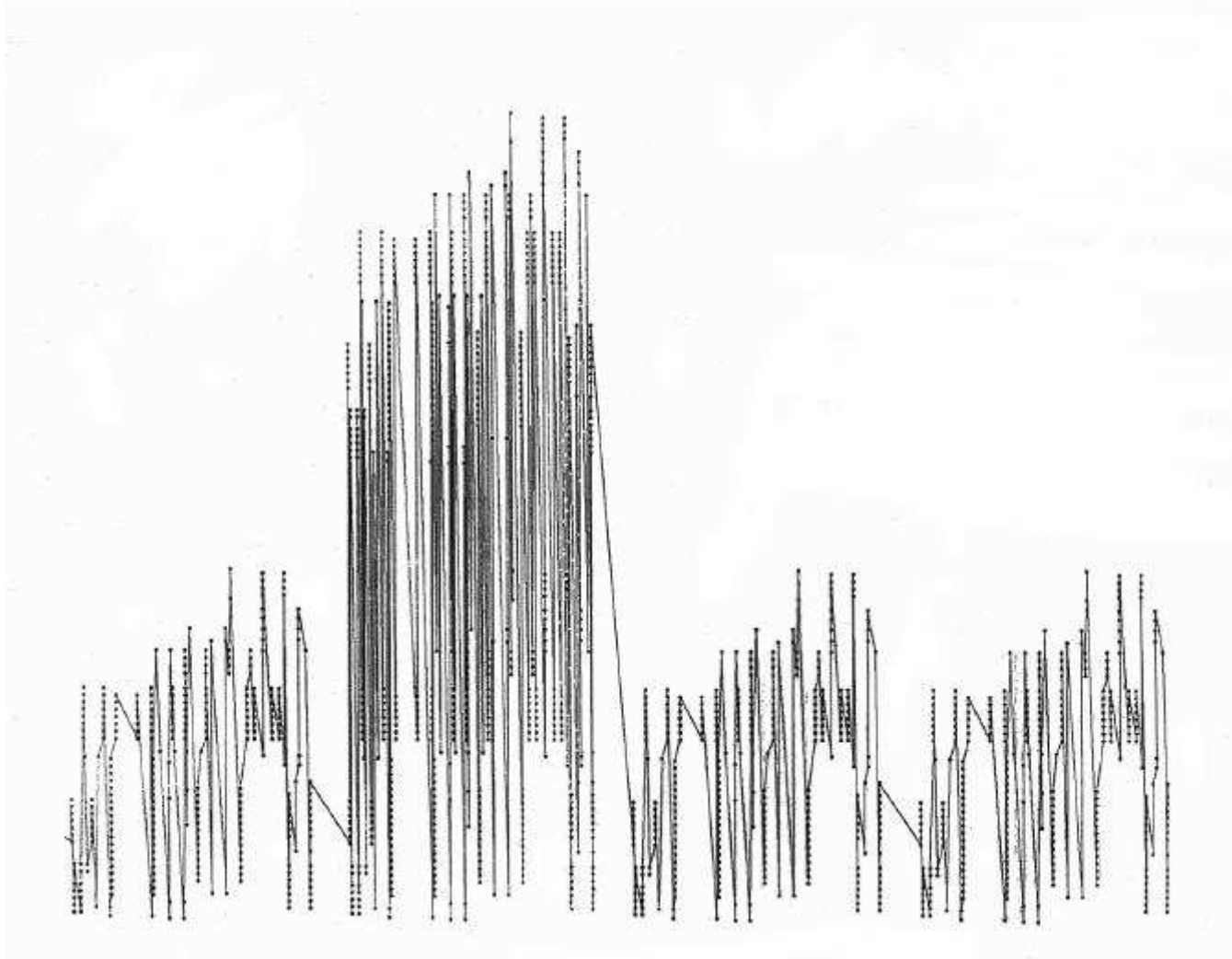
Feasibility heuristic

- Adjusts Lagrangian solution \mathbf{x} such that the resulting vector is feasible
- Often a good thing to do when approaching the dual optimal solution— \mathbf{x} often then only mildly infeasible
- Identify path in 1-MST with many links; form a subgraph with the remaining nodes which is a path; connect the two
- Result: A Hamiltonian cycle (TSP tour)
- We then have both an upper bound (feasible point) and a lower bound (q) on the optimal value—a quality measure: $[f(\mathbf{x}) - q(\boldsymbol{\mu})]/q(\boldsymbol{\mu})$

The Philips example

- Fixed number of subgradient iterations
- Feasibility heuristic used every K iterations ($K > 1$), starting at a late subgradient iteration
- Typical example: Optimal path length in the order of 2 meters; upper and lower bounds produced concluded that the relative error in the production plan is *less than 7 %*
- Also: increase in production by some 70 %

Initial drill pattern



Near-optimal drill pattern



A research topic at MV: Opportunistic maintenance optimization

- What? When performing maintenance, use the opportunity to replace more parts than necessary for the sake of an overall best maintenance plan
- Why? Law (safety), economics, customer relations, . . .
- Where? Any production facilities with large fixed maintenance costs (expensive shut downs) — steel/aluminium, paper, plastic, power production/distribution

- How?
 - Data: remaining lives of components, predictions of wear, safety requirements, costs of new parts, work costs
 - Constraints: maintain within the life limits (not too late); minimum condition of system at the end of the time period considered; safety/redundancy . . .
 - Objective: minimize overall running and maintenance costs

Example: Volvo Aero, Trollhättan

- Maintenance of the RM12 engine (JAS and civil aircraft)
- The engine consists of several modules; parts in modules are either safety-critical (typically rotating ones) or on-condition
- Safety-critical parts have fixed lives (deterministic); also others are monitored and are considered stochastic—conditional lives
- Goal: Maintain the whole fleet such that the total maintenance cost is minimized (or the total “time on wing” is maximized)

- Data: costs for inspection, (dis)assembly, cost of parts, lives of new and old parts ...
- Data for simple model example:
 - $z_t \in \{0, 1\}$: perform maintenance or not at time t
 - $x_{it} \in \{0, 1\}$: replace part $i = 1, \dots, N$ or not at time t
 - T_i : life of part i (if new)
 - $t = 1, \dots, T$: time discretization up to the horizon
 - c_i : cost of part i (if new)
 - d : fixed cost for performing maintenance
- Often one has used parts (life is $\tau_i < T_i$)

$$\begin{aligned}
&\text{minimize} && \sum_{t=1}^T \left(\sum_{i \in \mathcal{N}} c_i x_{it} + dz_t \right) \\
&\text{subject to} && \sum_{t=\ell}^{T_i+\ell-1} x_{it} \geq 1, \quad \ell = 1, \dots, T - T_i, \quad i \in \mathcal{N} \\
&&& x_{it} \leq z_t, \quad t = 1, \dots, T, \quad i \in \mathcal{N} \\
&&& x_{it}, z_t \in \{0, 1\}, \quad t = 1, \dots, T, \quad i \in \mathcal{N}
\end{aligned}$$

- Replace within each component's life time
- If replacement is done, the fixed cost must be paid
- In fact the requirement $z_t \in \{0, 1\}$ can be replaced by $z_t \in [0, 1]$ —it will be integral at an optimal solution!

- Let

$$T = 60, \quad N = 4,$$

$$T_1 = 13, \quad T_2 = 19, \quad T_3 = 34, \quad T_4 = 18,$$

$$c_1 = 80, \quad c_2 = 185, \quad c_3 = 160, \quad c_4 = 125$$

- Figure 1 shows maintenance occasions for three cases of fixed costs (the second is the most realistic)
- Each maintenance occasion is represented by a vertical bar, where the presence of a colored dot at a given height 1–4 represents an item being replaced
- The figure clearly illustrates how opportunistic maintenance becomes more beneficial with an increase in work costs

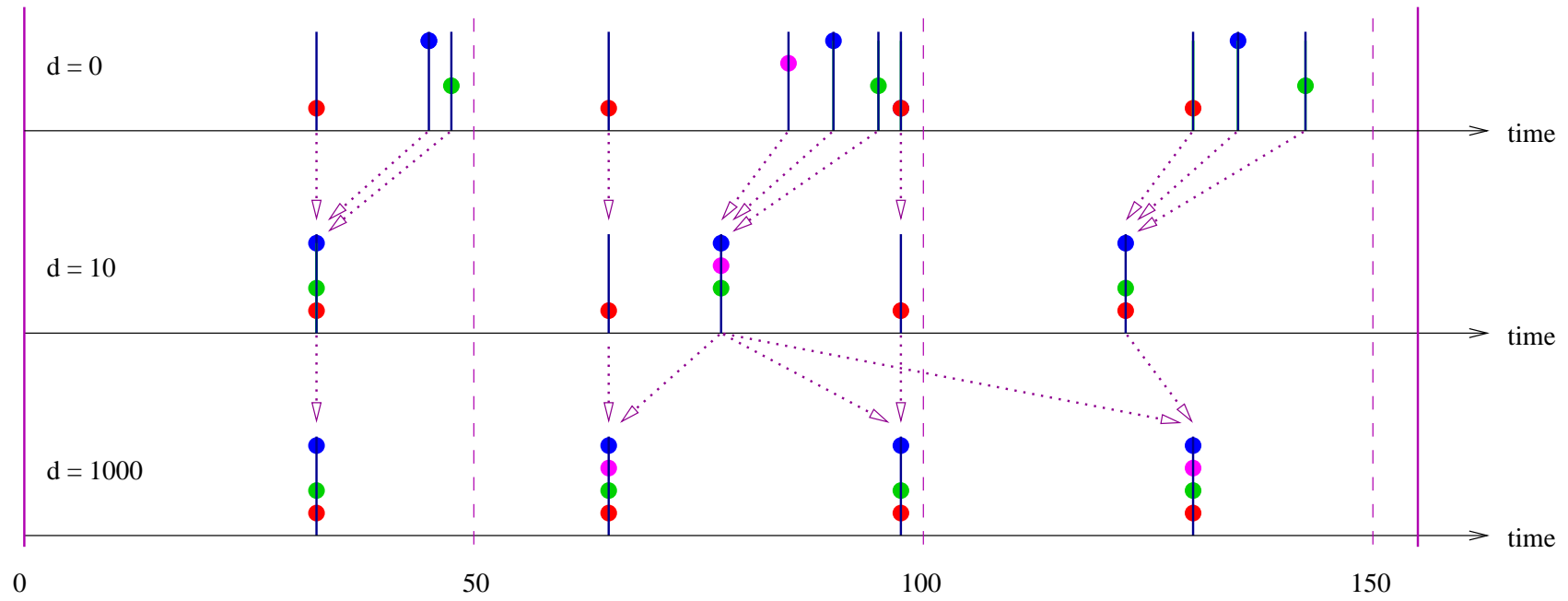


Figure 1: Optimal solutions for three cases of fixed costs

Status and goals for the future

- Current status:
 - System may be implemented at Volvo Aero
 - Collaborations: Electrical Engineering, KTH & Energy and Environment, Chalmers (nuclear/wind power production)
- Create optimization model and solution methodology system for general cases and systems
- Recent research grant for one PhD student; more to come
- Searching for masters students!
- We have a maintenance workshop at Chalmers in December