Optimization for sustainable development

Can OR truly help ?

OSD Chair

CNRS LIX, École Polytechnique

ROADEF, Toulouse, 2010

Outline

Sustainable Development

Sustainable Development is a paradigm for natural resources utilization that aims

- at meeting human needs
- while preserving the environment
- so that these needs can be met not only in the present, but also for future generations.

Sustainable development offers a vision of progress that integrates immediate and longer-term objectives, local and global actions, and has a profound beneficial impact on social, economic and environmental issues, viewed as inseparable and interdependent components of human progress.

Project / Partners

Key sustainability issues translate into decision and optimization problems

- Constraint Reasoning Group (Microsoft Research)
- OR group of the CS lab of Ecole Polytechnique / CNRS
- Launched in June 2009 with Philippe Baptiste, Christoph Durr, Youssef Hamadi, Vincent Jost, David Savourey, Nora Touati.

Related projects : Institute for Computational Sustainability (Cornell : Gomes, Shmoys, Conrad, Hopcroft, Selman, ...),

Can we help ?

Core Competencies

- Modeling
  - Stochastic models
  - Multi-Objective Optimization
- Problem Solving Techniques
- Well defined methodology

Global *vs.* local

- We can solve small local and well-defined problems
- What's our real impact ?

Outline

Server Farms

- The Environmental Protection Agency expects U.S. data centers to consume more than 100 billion Kilowatt-hours by next year
- about the same amount of electricity drown by 11 million U.S. households
- Huge waste to cope with peak processing demands (30% utilization rate).

Huge amount of research on how to reduce energy. We focus on power management policies of CPUs.

Power Management policies

Basic mechanisms

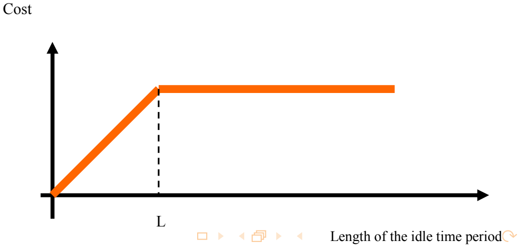
- *Speed scaling*: frequency of processors can be changed online
  - Energy required to perform the same task increases with the speed of the processor,
  - Speed scaling policies slow down the processor as much as possible
- *Shut-down*: put a system into a sleep state when it is idle.
  - Small amount of energy consumed in the sleep state but
  - a fixed amount of energy is required when resuming

More complex systems play with several sleep states+ speed scaling techniques (see Sandy Irani and Kirk Pruhs up-to-date survey on algorithmic problems in power management)

Model

- n tasks $1, \ldots, n$ are to be processed on the processor
- Each task i is associated with a release date r_i, a deadline d_i and a processing time p_i.
- Tasks can be interrupted at any integer time point
- The energy required by a transition from the sleep state to the active state is L times the energy required over a unit time slot (in the active state)
- Objective = minimize total energy required

We look for a single machine preemptive schedule with *few and long idle time periods* (When L is 1 = schedule with a minimum number of idle time intervals)



Model

- Complexity status open (even for $\forall i, p_i = 1$)
  - The problem of deciding if there is a schedule with no idle time interval is easy
- "Many seemingly more complicated problems in this area can be essentially reduced to this problem, so a polynomial time algorithm for this problem would have wide application." (I & P, 05)
- Good news : the problem is polynomial for unit processing times
  - Dominance properties + Dynamic programming
- Bad news : $O(n^7)$

From now on, focus on the unit processing time problem

Simple ideas that do not work

Preemptive scheduling with time windows :

- Dispatching rules
- Assignment problem (assign jobs to time slots)

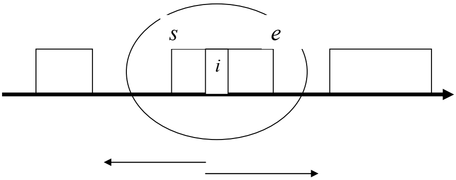
In both case, no straightforward extension

Outline

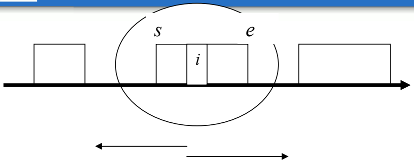
Dominance Properties

There is an optimal schedule in which, for any task i, the distance between the starting time of i and one of the release dates or deadlines is at most n

- $\mathcal{S}$ = optimal schedule that lexicographically minimizes the vector of starting times
- Let i be a task such that $\forall j, |t_i - r_j| > n$ and $\forall j, |t_i - d_j| > n$.
- Consider the largest "block" $[s, e]$ contain. i with no idle time



Dominance Properties



- Build 2 schedules by moving all tasks in $[s, e)$ from 1 unit either to the right (schedule $\mathcal{R}$) or to the left (schedule $\mathcal{L}$)
  - Valid because we are far away from release dates and deadlines
- $\mathcal{S}$ is optimal and lexic. minimal $\rightarrow$ $\mathcal{L}$ is strictly worse than $\mathcal{S}$.
- Hence, the idle period before $[s, e)$ on $\mathcal{S}$ is larger than L while the idle period after $[s, e)$ on $\mathcal{S}$ is smaller than L
- Hence the schedule $\mathcal{R}$ strictly improves on $\mathcal{S}$. Contradiction

Dominance Properties

There is an optimal schedule in which, for any task i, the distance between the starting time of i and one of the release dates or deadlines is at most n

$\Rightarrow$ we know that there is an optimal schedule in which start and completion times belong to the set Θ.

$$\Theta = \bigcup_i \{r_i - n, ..., r_i + n\} \cup \{d_i - n, ..., d_i + n\}$$

Good news : there is a quadratic number of time points in Θ.

Outline

Decomposition

- Jobs are sorted in non decreasing order of deadlines, i.e., $d_1 \leq d_2 ... \leq d_n$.
- If job n is scheduled at time t then we can assume that there is no other job available at t (*i.e.*, if $r_u \leq t$ then u is already scheduled)
  - Otherwise swap u and i (works fine because $u < n \Rightarrow d_u \leq d_n$)
- This allows us to decompose the problem
  - Before time t with $\{u < n : r_u \leq t\}$
  - After time $t + 1$ with $\{u < n : r_u > t\}$
- Our algorithm applies recursively this decomposition

Decomposition

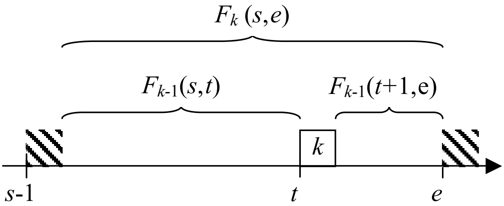
- Definition: *relative cost of a given schedule over an interval* $[s, e)$ = cost of the schedule built as follows: schedule a fake job in $[s-1, s)$, follow the initial schedule from s to e and finally schedule a fake job in $[e, e+1)$
- For any integer $k \leq n$, let $F_k(s, e)$ be the minimal relative cost over the interval $[s, e)$ among all schedules of the jobs $\{i \leq k, s \leq r_i < e\}$ such that
  1. the machine is idle before s and after e,
  2. starting times and completion times are in Θ.

  If no such schedule exists, $F_k(s, e) = \infty$.
- the optimum $= F_n(\min \Theta - L, \max \Theta + L) - 2L$ (the "$-2L$" takes into account fake jobs that always create two interruptions with cost L)
- We have $F_0(s, e) = \min(L, e - s)$

Decomposition

- If job $k > 0$ is such that $r_k \notin [s, e)$ then $F_k(s, e) = F_{k-1}(s, e)$
- If $r_k \in [s, e)$ then

$$F_k(s, e) = \min_{t, t+1 \in \Theta, s \leq t < e} F_{k-1}(s, t) + F_{k-1}(t + 1, e)$$



Outline

A Dynamic Program

Optimisation for Sustainable Development

- The relevant values for s and e are exactly those in Θ
- The values of $F_k(s, e)$ are stored in a multi-dimensional array of size $O(n^5)$ (n possible values for k, n^2 possible values both for s and e)
- Initialization : $F_0(s, e)$ is set to $\min(L, e - s)$ for any values s, e in Θ ($s \leq e$).
- Iterate from $k = 1$ to $k = n$. Each time, F_k is computed for all the possible values of the parameters thanks to the Decomposition Proposition and to the values of F_{k-1} computed at the previous step.

A Dynamic Program

Complexity:

- Initialization phase runs in $O(n^4)$
- For each value of k, $O(n^4)$ values of $F_k(s, e)$ have to be computed. For each of them, a maximum among $O(n^2)$ terms is computed
- Leads to an overall time complexity of $O(n^7)$
- Space complexity $= O(n^5)$

Extended results

- Generalization : task i has an arbitrary processing time $p_i \in \mathbb{N}$ (+ preemption)

	$p_i = 1$	general p_i
$L = 1$	$O(n^7) \rightarrow O(n^4)$	$O(n^5)$
general L [BCD 08]	$O(n^4)$	$O(n^5)$

- Parallel machines [E. Demaine 08, A. Lopez 08]

Is it a useful result ?

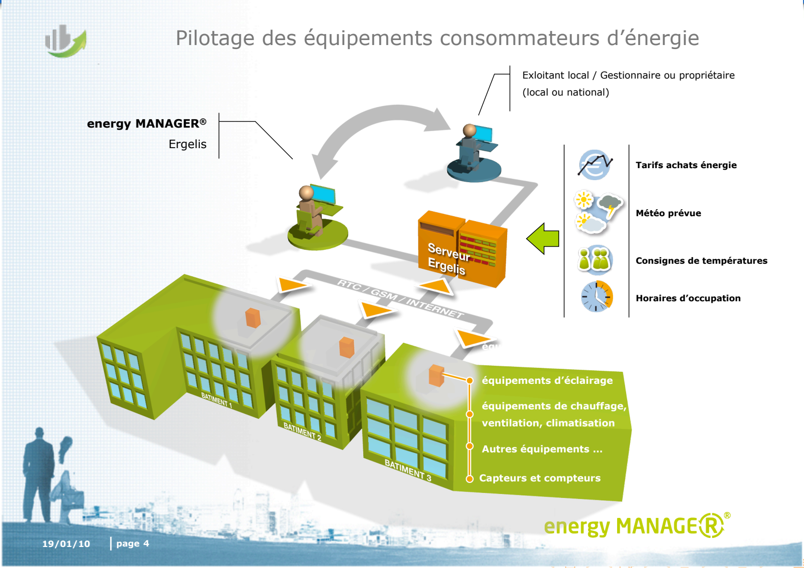
Outline

Reducing energy consumption of building

- Temperature regulation policies of buildings rely on basic mechanisms :
  - set of independant thermostat + heating and cooling devices
  - slightly better : interconnected devices connected to the internet ("internet managed energy control")
- Not very efficient in practice

Reducing energy consumption of building

Basic Model ©Ergelis

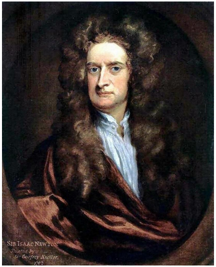


Basic Model

- Weather forecast (over time)
- Required temerature windows (over time)
- Number of people in the building + machines (over time)
- Basic model of the building
- Energy cost over time (electricity, gaz, *etc.*)

Objective : schedule all heating/cooling devices to minimize total cost

Basic Model, Newton's law of cooling



$$\frac{dT}{dt} = hA(T_e(t) - T(t)) + H(t)$$

- Q = Thermal energy
- h = Heat transfer coefficient
- A = Surface of the transfer area
- T = Temperature of the building
- T_e = External temperature
- $H(t)$ = heating rate

How to solve the problem ?

- Production planning problem
- Time is discretized
- decision variables = amount of energy spent on a given device during a given time-slot
- Room temperatures can be computed according to a discretized version of the building model (take into account external temperature)
- Small Mixed Integer Program

Results

- MIP is most often solved to optimality
- The rough model used is good enough
- Up to 30% saving
- Issue : air renewal regulation, ...

Is it useful ?

Outline

Road transport

- Road transport is predicted in the EC to grow by a further 33% in the next 20 years (www.europa.eu (accessed 13/01/2007)).
- Transport areas (particularly road and air travel) face today the challenge of making transport sustainable in
  - environmental (CO2, air pollution, noise).
  - competitiveness (congestion).
  - addressing social concerns.

Outline

Fast computation of shortest paths: why

- Route planners in road networks need *fast* algorithms (e.g. Google Maps)
- Problem: given a digraph $G = (V, A)$, compute shortest path between two nodes $s, t \in V$
- Could use Dijkstra's algorithm (polynomial time), but it is too slow (several seconds for each computation) on large graph: European road network has $\approx$ 20M nodes, 40M arcs
- Need speedup techniques
- We study a more difficult version of the problem: the graph is *time-dependent* (arc cost functions which give travel time over an arc for each time of the day) and *dynamic* (the arc cost functions can be updated)

Propagation of dynamic information

- Various kind of sensors (cams, electromagnetic loops) retrieve real-time traffic information on important road segments such as motorways, etc.
- We have a sufficiently accurate estimation of the travel time over a subset of arcs S; this information is updated every few minutes
- We use a propagation algorithm to extend this information to a larger subset of arcs $S' \supseteq S$; the algorithm can be iterated to enlarge S'
- Idea, based on empirical knowledge: if we know that there is a congestion over two non-adjacent arcs which are near to each other (e.g. one or two hops), there is a large probability that the arcs between those two will also be congested

Solution method

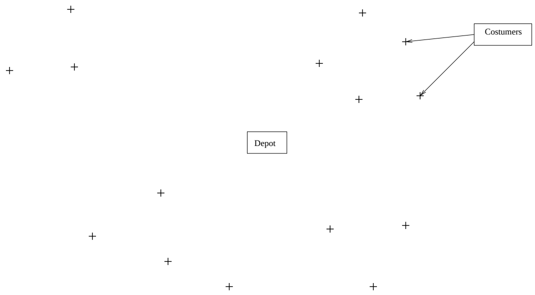
- Offline preprocessing stage (takes time, done only once) on the input graph to compute useful data, online path computation phase (accelerated using the preprocessed data)
- We developed a bidirectional, hierarchical search algorithm based on A^*, specific for time-dependent networks
- When the arc cost functions are updated with real-time traffic information, we need to update the preprocessed data (can do in ≈ 1 second)
- Speedup over Dijkstra of 3 orders of magnitude (both computing time and size of the search space): on the European road network, our average computing time for a shortest path is ≈ 10 milliseconds

Outline

1. OSD
2. On scheduling computer systems
3. Energy Control of Buildings
4. Green Vehicle Routing
   - Fast Shortest Path
   - VRSP : Time-dependent version
   - VRSP : Real-time version
   - Hazardous Materials
5. Waste Management and Handling

Vehicle Routing and Scheduling (VRSP)

Build an optimal set of routes for fleet of vehicles in order to serve a given set of customers.

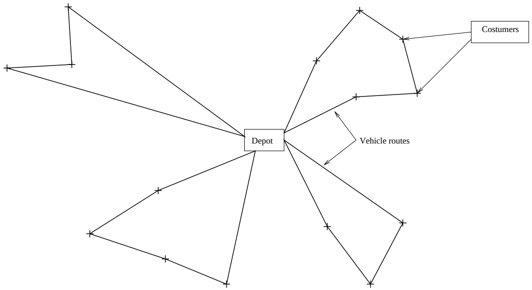


Many variants of the VRP

Capacitated VRP, VRP with Time Windows, VRP with Pick-Up and Delivering, Multiple Depot VRP

Vehicle Routing and Scheduling (VRSP)

Build an optimal set of routes for fleet of vehicles in order to serve a given set of customers.

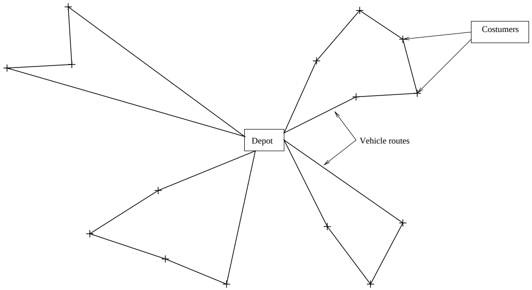


Many variants of the VRP

Capacitated VRP, VRP with Time Windows, VRP with Pick-Up and Delivering, Multiple Depot VRP

Vehicle Routing and Scheduling (VRSP)

Build an optimal set of routes for fleet of vehicles in order to serve a given set of customers.



Many variants of the VRP

Capacitated VRP, VRP with Time Windows, VRP with Pick-Up and Delivering, Multiple Depot VRP

Time-dependent VRSP

Problem description

- Dynamic travel and service time.
- Traffic conditions (peak times, traffic congestion).
  - Minimizing the total traveling time can lead to congestion.
  - Greater total travel distance can imply better speeds.
- Data requirements are significantly higher than for conventional models.

Outline

Real-time VRSP

Problem description

- Developed to assist in the real-time management and control of a distribution operation.
- Potentials for environmental benefits: Redirection to avoid unexpected congestion (accident).
- Models requiring real-time informations on the location of vehicles and current traffic conditions.
- Differences from the conventional VRSP: Time dimension, Probabilistic informations, Information update mechanisms, Computation time, ...

Environment issues

Dynamic VRSP, Time-dependent VRSP and environmental issues

- Some VRSP consider alternative objectives to pure economic considerations.
- Time-dependent VRSP: produce indirectly less pollution.
- No models where environment objectives (in terms of a measure of pollution) are considered.

Outline

Transport of hazardous materials

- Transportation of explosives, gazes, radioactive materials, flammable liquids and solids, ...
- An important problem in industrialized societies.
- A significant role of OR for hazmat transport.
- Two research categories:
  - Risk analysis
  - Routing/scheduling

Transport of hazardous materials

Model to asses risk

- Consider a path r consisting of the set of links $\{l_1, ..., l_q\}$.
- Each link consists of two important attributes:
  - $p(l_i)$, the probability of a release accident on link i.
  - $C(l_i)$, a measure of the consequence of a release accident on link l_i (exp. the number of people living within a mile).
- $TR(r) = \sum_{l_i \in r} p(l_i)C(l_i)$.

Transport of hazardous materials

Model to asses risk

- Traditional risk (probability of incident $\times$ consequence of incident).
- Population exposure (total number of people exposed to risk during a transport activity).
- Incident probability.
- Perceived risk ($PR(r) = \sum_{I_i \in r} p(I_i) C^q(I_i)$).

- Very intuitive models
- but it is hard to evaluate accidents frequency and their consequences (scarcity of data, variability of the conditions along the routes).

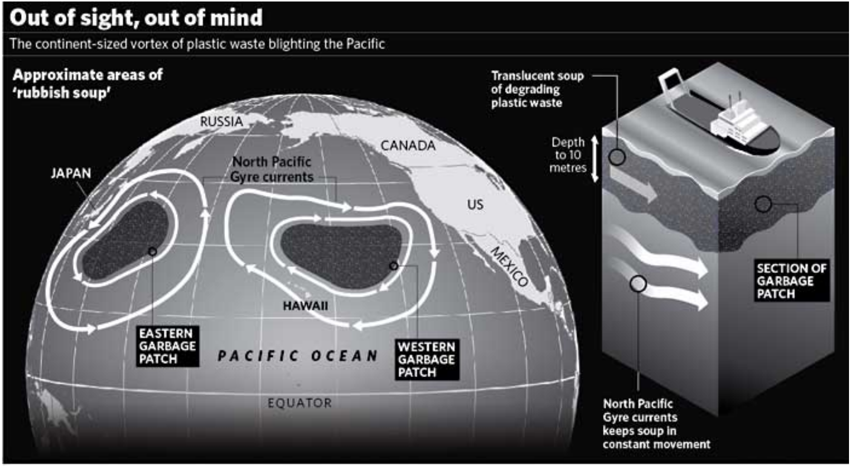
Some resolution methods

Multi-objective models

- Column generation and set partitioning approach [1].
- Branch-and-bound method [2].
- Lagrangian relaxation [3].
- Dynamic programming techniques [4].

Outline

Trash Vortex



Source: Greenpeace

Graphic: John Papasian, John Bradley

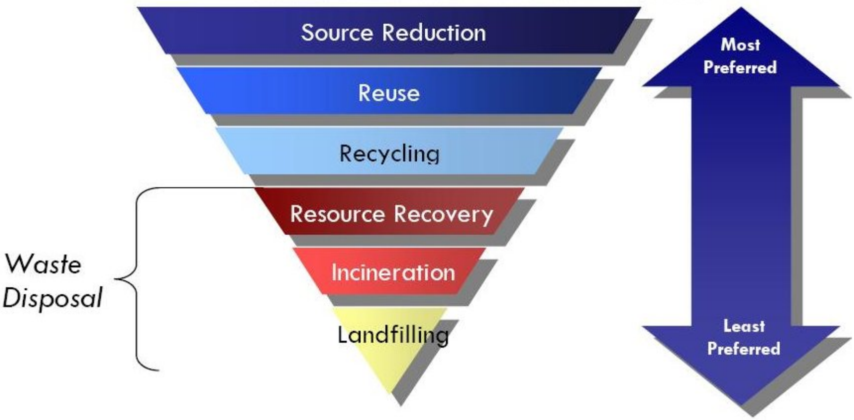
Definition of waste

- "Any material thing whose owner wants to get rid of"
- "Any residue whose appropriate treatment can benefit human or environmental health"

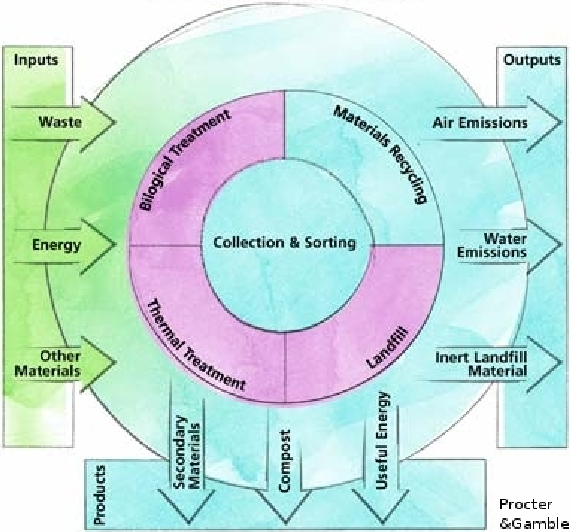
Definition of waste

- "Any material thing whose owner wants to get rid of"
- "Any residue whose appropriate treatment can benefit human or environmental health"

Waste hierarchy vs. Decision Making



Waste hierarchy vs. Decision Making



Procter &Gamble

Decision Making

Multi-objective evaluation/optimization (economical, environmental and social)

Models for choosing and coordinating macro-decisions

- Presort and store to avoids biochemical deterioration STOCK
- Collect VRP
- Sort : separation networks (human and mechanical) FLOW
- Treat (Wash, Compost, Incinerate, Landfill. . .) ASSIGNMENT
- Recycle : depends on the quality of obtained waste INPUT
- Secondary material : depends on industrial network OUTPUT

Sorting Municipal Waste

- Picture: AMB

What are we (OR) doing ?

Many papers dedicated to specific problems

- Collecting as constrained VRP
- Choice of Treatment as a multi-objective assignment problem
- Facility location of landfill and sorting factory
- Composting and incineration as functions of waste composition
- Empirical analysis, statistics and optim models for sub-problems

Very few global approaches

Todo list

Global view

From marketing to recycling via Ecodesign

Model objective functions

For a company/a citizen, “polluting” is still cheap and by-products of ecosystems have limited financial value (pollination, drinking water, fish stocks)

- Consequences of pollution on human health are not clear
- Consequences of pollution on nature are not clear

Outline

What we have seen

- Very early step of the project : list of relevant problems that can be adressed by OR
- Interaction required with economist, ecologist, ...
- OR can help to model (and solve) problems related so SD but
  - Objectives are not clearly stated
  - “Cost” functions are very difficult to set up
  - “Global” view is required to be efficient
  - Policy-maker want advices (and not complex software)

Wake effects at offshore wind farm

- Reduced power, increased fatigue loading on downstream turbines + increased noise from downstream turbines
- Given *n* turbines, a patch of land and a distribution over wind direction, what is the "optimal" placement? One would presumably want to minimize the expected shadowing under your distribution.

Next assignment for you students (found yesterday on `withouthotair.blogspot.com`, thanks to Fabien Petitcolas)

- Raj A. Sivakumar, Rajan Batta and Mark H. Karwan, A multiple route conditional risk model for transporting hazardous materials, INFOR, 20-33, 1995.
- H.D. Sherali, L.D. Brizendine, T.S. Glickman, and S. Subramanian, Low probability-high consequence considerations in routing hazardous material shipments. Transportation Science, 31:237-251, 1997.
- Raj A. Sivakumar and Rajan Batta, The variance constrained shortest path problem. Transportation Science, 28:309-316, 1994.
- D.A. Nembhard and C.C. White, Applications of non-order-preserving path selection to hazmat routing. Transportation Science, 31:262-271, 1997.