Additive Bounding, Dual Ascent and Exact Algorithms applied on set partitioning-like formulations

Roberto Wolfler Calvo

LIPN - AOC, Université Paris 13

14 Mars 2014

Outline

Outline

Set partitioning: problem definition

- Let $M = \{1, \ldots, m\}$ a set of m objects
- Let $N = \{1, \ldots, n\}$ be the index set of n subsets $R_1, \ldots R_n$ of M
- Each subset R_j has an associated cost c_j

The Set Partitioning is the problem of finding a minimum family of subsets R_j, $j \in N$ which is a partition of M

Set partitioning: mathematical formulation

The mathematical formulation of the set partitioning problem is the following

$$\min z(P) = \sum_{j \in N} c_j x_j \tag{1}$$

$$s.t. \sum_{j \in N_i} x_j = 1, \quad \forall i \in M \tag{2}$$

$$x_j \in \{0, 1\} \quad \forall j \in N \tag{3}$$

where $N_i \subseteq N$ is the index set of the subset covering the object (row) $i \in M$ (i.e., $N_i = \{j \in N : i \in R_j\}$)

Set partitioning: mathematical formulation (2)

The dual D of the LP relaxation of P is

$$\max z(D) = \sum_{i \in M} u_i \tag{4}$$

$$s.t. \sum_{i \in R_j} u_i \leq c_j, \quad j \in N \tag{5}$$

$$u_i \in \mathbb{R}, \quad i \in M \tag{6}$$

The dual ascent heuristic is based on a parametric relaxation based on the following parametric variable substitution.

variable substitution

Let us introduce

- A positive weight q_i
- $q(R_j) = \sum_{i \in R_j} q_i$
- A set of variable y_j^i which is equal 1 if row $i \in R_j$ is covered by column j

Then, a variable substitution is possible

$$x_j = \sum_{i \in R_j} \frac{q_i}{q(R_j)} y_j^i \quad j \in N \tag{7}$$

The expression (7) imposes that if $x_j = 1$ then $y_j^i = 1$, for each $i \in R_j$, while if $x_j = 0$ then $y_j^i = 0$, for each $i \in R_j$

The parametric set partitioning

The mathematical formulation of the relaxation when we use 7 becomes

$$z(RP(q)) = \min \sum_{j \in N} c_j \sum_{i \in R_j} \frac{q_i}{q(R_j)} y_j^i \tag{8}$$

$$s.t. \sum_{j \in N_i} \sum_{k \in R_j} \frac{q_k}{q(R_j)} y_j^k = 1, \quad \forall i \in M \tag{9}$$

$$\sum_{j \in N_i} y_j^i = 1, \quad \forall i \in M \tag{10}$$

$$y_j^i \in \{0,1\} \quad \forall i \in M, j \in N \tag{11}$$

Note that the feasible solution set of $RP(q)$ contains the feasible solution set of P, since (7) transforms any solution of P into a $RP(q)$ solution and also a $RP(q)$ solution into a fractional solution of P

The linear relaxation of RP

The mathematical formulation of the lagrangian relaxation of RP is

$$z(LRP(q,\lambda)) = \min \sum_{i \in M} \Big(\sum_{j \in N_i} (c_j - \lambda(R_j)) \frac{q_i}{q(R_j)} y_j^i + \lambda_i \Big) \tag{12}$$

$$s.t. \ \sum_{j \in N_i} y_j^i = 1, \quad \forall i \in M \tag{13}$$

$$y_j^i \in \{0,1\} \quad \forall i \in M, j \in N \tag{14}$$

where $\lambda(R_j) = \sum_{i \in R_j} \lambda_i$

The linear relaxation of RP (2)

Problem $LRP(q, \lambda)$ is decomposable into M subproblems, one for each object $i \in M$, and can be solved by inspection as follows

- Let $j_i \in N_i$ the index of the column covering row $i \in M$ such that

$$\frac{q_i(c_{j_i} - \lambda(R_{j_i}))}{q(R_{j_i})} = \min_{j \in N_i} \frac{q_i(c_j - \lambda(R_j))}{q(R_j)}$$

- Then the optimal solution y of problem $LRP(q, \lambda)$ is obtained by setting
  - $y^i_{j_i} = 1$ and $y^i_j = 0 \quad \forall j \in N_i \setminus \{j_i\}, i \in M$
- The cost $z(LRP(q, \lambda))$ of the optimal solution is the following
  - $z(LRP(q, \lambda)) = \sum_{i \in M} \frac{q_i(c_{j_i} - \lambda(R_{j_i}))}{q(R_{j_i})}$

The linear relaxation of RP (3)

Theorem

Any optimal $LRP(q, \lambda)$ *solution for a given vector* $\lambda \in \mathbb{R}^m$ *and* $q \geq 0$ *provides a feasible solution* u *of the dual problem* D *of cost* $Z_D(\lambda, q) = z(LRP(\lambda, q))$ *that is given by the following expression*

$$u_i = \frac{q_i(c_{j_i} - \lambda(R_{j_i}))}{q(R_{j_i})} + \lambda_i \quad i \in M$$

The proof is based on the idea that the inequalities

$$\sum_{i \in R_j} u_i \leq c_j$$

are respected

The linear relaxation of RP (4)

Proof.

Note that

$$\frac{q_i(c_{j_i} - \lambda(R_{j_i}))}{q(R_{j_i})} \leq \frac{q_i(c_j - \lambda(R_j))}{q(R_j)} \quad \forall i \in R_j$$

but also

$$u_i \leq \frac{q_i(c_j - \lambda(R_j))}{q(R_j)} + \lambda_i \quad \forall i \in R_j$$

and if we sum up for all $i \in R_j$

$$\sum_{i \in R_j} u_i \leq \sum_{i \in R_j} \frac{q_i(c_j - \lambda(R_j))}{q(R_j)} + \sum_{i \in R_j} \lambda_i = c_j$$

□

Dual value

Corollary

For every pair of vectors $\lambda \in \mathbb{R}^m$ *and* $q > 0$ *The following inequalities hold:*

$$z(LRP(\lambda, q)) \leq Z_D^*$$

Proof.

If follows directly from Theorem 1 □

Z(LRP) .vs. Z(CLR)

Theorem

The following inequalities hold:

$$z(LRP(\lambda, q)) \geq Z(CLR(\lambda)) \quad \forall \lambda \in \mathbb{R}^m, \forall q > 0$$

Moreover the inequality becomes strict if there exists at least one row i that satisfies the following inequality

$$c_j - \lambda(R_j) > 0 \quad \forall j \in N_i$$

Z(LRP) .vs. Z(CLR) (2)

Proof.

Using the solution y of problem $LRP(\lambda, q)$ define the variable $x_j, j \in N$ according to expression (7). Let $J = \{j \in N : x_j > 0\}$ and
$\tilde{N} = \{j \in J : c_j - \lambda(R_j) < 0\}$
Then the cost $z(LRP(\lambda, q))$ of the optimal solution y is:

$$z(LRP(\lambda, q)) = \sum_{j \in \tilde{N}} (c_j - \lambda(R_j))x_j + \sum_{j \in J \setminus \tilde{N}} (c_j - \lambda(R_j))x_j + \sum_{i \in M} \lambda_i$$

since $\sum_{j \in J \setminus \tilde{N}} (c_j - \lambda(R_j))x_j \geq 0$ we can have

$$z(LRP(\lambda, q)) \geq \sum_{j \in \tilde{N}} (c_j - \lambda(R_j))x_j + \sum_{i \in M} \lambda_i$$

As $x_j \leq 1, j \in J$ and $\tilde{N} \subseteq \bar{N}$ we have $\sum_{j \in \tilde{N}} (c_j - \lambda(R_j))x_j \geq \sum_{j \in \bar{N}} (c_j - \lambda(R_j))$ and therefore

$$z(LRP(\lambda, q)) \geq \sum_{j \in \bar{N}} (c_j - \lambda(R_j)) + \sum_{i \in M} \lambda_i = z(CLR(\lambda))$$

Example of Z(LRP) .vs. Z(CLR)

$m = 7$, $n = 9$, $c = (2, 2, 1, 1, 1, 6, 7, 5, 6)$ and $\lambda = (1, 1, 1, 1, 0, 0, 0)$ and $q = (1, 1, 1, 1, 1, 1, 1)$ and the coefficient matrix is

$$A = \begin{bmatrix} 1 & & & & & 1 & 1 & & \\ 1 & & & & & & 1 & 1 & \\ & 1 & & & & 1 & & 1 & 1 \\ & 1 & & & & & 1 & & 1 \\ & & 1 & & & & 1 & & \\ & & & 1 & & 1 & & & 1 \\ & & & & 1 & 1 & & 1 & \end{bmatrix}$$

Then we can calculate $c' = (0, 0, 1, 1, 1, 4, 4, 3, 4)$ and therefore $z(CLR(\lambda)) = \sum_{i \in M} \lambda_i = 4$ while $z(LRP(\lambda, q)) = y_3^5 + y_4^6 + y_5^7 + \sum_{i=1}^{7} \lambda_i = 7$

Improving the Lower Bound $z(LRP(\lambda, q))$

The lower bound $z(LRP(\lambda, q))$ can be improved if there eists a row $i \in M$ where every column $j \in N_i$ has a strictly positive reduced cost $c_j - \sum_{i \in R_j} u_i$ by increasing the dual variable u_i.
It can be increased, for example, by setting $\lambda^{t+1} = u^t$

The maximum value of $z(LRP(\lambda, q))$

Corollary

The following inequalities hold:

$$\max_{\lambda,q} z(LRP(\lambda,q)) = \max_{\lambda} z(LRP(\lambda,q')) = Z_D^* \quad \forall q' > 0$$

Proof.

If follows directly from Theorem 1 and 3 and the well-known result that the Lagrangian dual in this case equals the LP value □

Outline

The additive bounding definition

Consider the following problem

$$\begin{aligned} \min z(P) = \sum cx \\ s.t.\ \ Ax = b, \\ x \geq 0, \quad integer \end{aligned}$$

The dual D of the LP relaxation of P is

$$\begin{aligned} \max z(D) = \sum wb \\ s.t.\ \ wA \leq c, \\ w, \quad unrestricted \end{aligned}$$

The additive bounding definition (2)

A feasible solution $\bar{w}$ of D of cost $\bar{z}(D)$ can be obtained as $\bar{z}(D) = z(D) + z'(D')$ and $\bar{w} = w + w'$. Where w is a feasible solution of D and w' is a feasible solution of the following problem D'

$$\max z'(D') = \sum w' b$$
$$s.t.\ \ w'A \leq c - wA,$$
$$w', \quad unrestricted$$

Note that D' impose $(w + w')A \leq c$
The additive bounding has been introduced by Fischetti and Toth in 1989

Outline

Set partitioning formulation for the CVRP

- Let $\mathscr{R}$ be the index set of all feasible routes and let $\mathscr{R}_i \subset \mathscr{R}$ be the index subset of the routes covering customer $i \in V_c$.
- Let $a_{i\ell}$ be a $(0-1)$ coefficient equal to 1 iff $i \in V$ belongs to route $\ell \in \mathscr{R}$.
- Each route $\ell \in \mathscr{R}$ has an associated cost c_ℓ. R_ℓ indicates the subset of vertices (i.e., $R_\ell = \{0, i_1, i_2, \ldots, i_h\}$) visited by route $\ell \in \mathscr{R}$.
- Let ξ_ℓ be a binary variable equal to 1 iff route $\ell \in \mathscr{R}$ is in solution.
- The Set partitioning formulation is the following:

$$\min \sum_{\ell \in \mathscr{R}} c_\ell \xi_\ell$$

$$s.t. \sum_{\ell \in \mathscr{R}} a_{i\ell} \xi_\ell = 1, \quad \forall i \in V_c, \tag{15}$$

$$\sum_{\ell \in \mathscr{R}} \xi_\ell = M, \tag{16}$$

$$\xi_\ell \in \{0, 1\}, \quad \forall \ell \in \mathscr{R}.$$

Set partitioning formulation for the CVRP (2)

- The set partitioning formulation remains valid if the set of routes $\mathscr{R}$ is enlarged with the set $\hat{\mathscr{R}} \supset \mathscr{R}$ containing, in addition to elementary routes, *non-elementary* routes (▸ Example), which are routes in which the vehicle is permitted to visit customers more than once.
- In this case, coefficient $a_{i\ell}$ is a general integer coefficient that is equal to the number of times customer i is visited by route ℓ.
- Note that the overall integer programming formulation remains valid, since constraints (15) ensures that the variables representing non-elementary routes will be automatically eliminated when ξ is binary.
- Although non-elementary routes are infeasible, this relaxation has the advantage that the pricing subproblem becomes solvable in pseudo-polynomial time.

Valid inequalities for the set partitioning formulation

- Any solution ξ of the set partitioning formulation can be transformed into a solution $\mathbf{x}$ of the two-index formulation by setting:

$$x_{ij} = \sum_{\ell \in \mathscr{R}} \eta_{ij}^{\ell} \xi_{\ell}, \quad \forall \{i,j\} \in E,$$

where the coefficients η_{ij}^{ℓ} are defined as follows:

- if ℓ is a single customer route covering customer h, then $\eta_{0h}^{\ell} = 2$ and $\eta_{ij}^{\ell} = 0$, $\forall \{i,j\} \in E \setminus \{0,h\}$;
- if ℓ is not a single customer route, then $\eta_{ij}^{\ell} = 1$ for each edge $\{i,j\} \in E(R_{\ell})$ and $\eta_{ij}^{\ell} = 0$, $\forall \{i,j\} \in E \setminus E(R_{\ell})$.

where $E(R_{\ell})$ represents the subset of the edge set E covered by route R_{ℓ}.

Solving the Pricing Problem

- The pricing problem consists of an Elementary Shortest Path Problem with Resource Constraints (ESPPRC).
- Let $\mathbf{u} = (u_0, u_1, \ldots, u_n)$ be the dual variables of (SP), where u_0 is associated with (16) and u_i, $i = 1, \ldots, n$, with (15).
- Given the reduced cost matrix $[\bar{d}_{ij}]$, where $\bar{d}_{ij} = d_{ij} - \frac{1}{2}(u_i + u_j)$, the ESPPRC calls for finding the cost of a least-cost route.

Solving the Pricing Problem
Exact Dynamic Programming Recursion

- Let $\mathcal{P}$ be the set of paths of G s.t. each path $P \in \mathcal{P}$ starts from 0, visits a set of vertices $V_P \subseteq V_c$, delivers q_P units of product, and ends at vertex $\sigma_P \in V_P$.
- The ESPPRC can be solved with Dynamic Programming (DP) recursions:
  - state-space graph $\mathcal{X} = \{(X, i) \ : \ X \subseteq V_c,\ i \in V\}$;
  - functions $f(X, i)$, $\forall (X, i) \in \mathcal{X}$, where $f(X, i)$ is the cost of a least-cost path P that visits the set of customers X, ends at customer $i \in X$, and such that $\sum_{j \in X} q_j \leq Q$.

Solving the Pricing Problem

q-route Relaxation

- [Christofides et al. 1981] proposed the *State-Space Relaxation* (SSR), that is a procedure whereby the state-space associated with a DP recursion is relaxed to compute valid bounds to the original problem.
- Elementary routes can be replaced with q-routes, which are nonnecessarily elementary routes delivering q units of product.
  - q-routes can contain loops.
  - 2-vertex loops can be easily avoided.
  - k-vertex loops (with $k \geq 3$) cannot be easily avoided.
- Given $[\bar{d}_{ij}]$, the cost of a least-cost q-route can be computed via DP in pseudo-polynomial time:
  - state-space graph $\mathcal{X} = \{(q,i) \ : \ i \in V, \, q_i \leq q \leq Q\}$;
  - functions $f(q,i)$, $\forall (q,i) \in \mathcal{X}$, where $f(q,i)$ is the cost of a least-cost path $P \in \mathcal{P}$ (nonnecessarily elementary) that ends at customer i and delivers q units of product.

Solving the Pricing Problem

ng-route Relaxation

- [Baldacci et al. 2011c] proposed the *ng*-route relaxation.
- For each path $P \in \mathcal{P}$, $P = \{0, i_1, \dots, i_{k-1}, i_k\}$, let P' be the path defined as $P' = \{0, i_1, \dots, i_{k-1}\}$.
- Let N_i ($N_i \subseteq V_c$) be a set of vertices associated with $i \in V_c$.
- With each path $P = \{0, i_1, \dots, i_k\}$, $P \in \mathcal{P}$, we associate the set $\Pi_P \subseteq V_P$ defined as: $\Pi_P = \{i_r \in V_{P'} \; : \; i_r \in \cap_{s=r+1}^{k} N_{i_s}\}$.
- Example:
    - $P = \{0, 1, 2, 3, 4, 1\} \Rightarrow P' = \{0, 1, 2, 3, 4\}$.
    - $N_1 = \{3, 4\}$, $N_2 = \{1, 5\}$, $N_3 = \{1, 4\}$, $N_4 = \{2, 3\}$.
    - $1 \notin N_2 \cap N_3 \cap N_4 \cap N_1$
      $2 \notin N_3 \cap N_4 \cap N_1$
      $3 \in N_4 \cap N_1$
      $4 \in N_1$
    - $\Rightarrow \Pi_P = \{3, 4\}$

Solving the Pricing Problem

The *ng*-route Relaxation

- An *ng*-path is a path $P \in \mathcal{P}$ s.t. $\sigma_P \notin \Pi_{P'}$ and P' is an *ng*-path.
- ... from the previous example:
  - $P = \{0, 1, 2, 3, 4, 1\} \Rightarrow P' = \{0, 1, 2, 3, 4\}$.
  - $N_1 = \{3, 4\}$, $N_2 = \{1, 5\}$, $N_3 = \{1, 4\}$, $N_4 = \{2, 3\}$.
  - $1 \notin N_2 \cap N_3 \cap N_4$
    $2 \notin N_3 \cap N_4$
    $3 \in N_4$
  - $\Rightarrow \Pi_{P'} = \{3\}$
  - $1 \notin \Pi_{P'}$ and P' is an ng-path (it is elementary!) $\Rightarrow P$ is an ng-path.
- An *ng*-route is an *ng*-path P plus the edge $\{\sigma_P, 0\}$.
- Given $[\bar{d}_{ij}]$, the cost of a least-cost *ng*-route can be computed with DP:
  - state-space graph $\mathcal{X} = \{(NG, q, i) \ : \ NG \subseteq N_i, \ i \in V, \ q_i \leq q \leq Q\}$;
  - functions $f(NG, q, i)$, $\forall (NG, q, i) \in \mathcal{X}$, where $f(NG, q, i)$ is the cost of a least-cost *ng*-path P that ends at customer i, delivers q units of product and s.t. $\Pi_P = NG$.

Solving the Master Problem

- The master problem is typically affected by degeneracy.
- Instead of using the simplex, we use a dual ascent heuristic relying on the following theorem:

Theorem 1.

Let $\lambda = (\lambda_0, \lambda_1, \ldots, \lambda_n)$ be a vector of penalties, where $\lambda_i \in \mathbb{R}$, $\forall i \in V_c$, are associated with (15) and $\lambda_0 \in \mathbb{R}$ with (16). A feasible dual solution $\mathbf{u}$ of cost $z(SP(\lambda)) = u_0 + \sum_{i \in V_c} u_i$ is obtained as:

$$\begin{cases} u_0 & = & \lambda_0, \\ u_i & = & q_i \min_{r \in \mathcal{R}} \left\{ a_{ir} \frac{c_r - \lambda_0 - \sum_{j \in V_c} a_{jr} \lambda_j}{\sum_{j \in V_c} a_{jr} q_j} \right\}, \quad \forall i \in V_c. \end{cases}$$

- A near-optimal dual solution of (SP) can be computed by mean of Theorem 1 and by applying subgradient optimization to update the penalty vector λ.

Adding Cuts from the (2I) to (SP)

- Any family of cuts valid for the $(2I)$ can be easily added to (SP).
- RCC (i.e., $\sum_{\{i,j\}\in\delta(S)} x_{ij} \geq 2k(S)$, $\forall S \in \mathcal{S}$), can be added as:

$$\sum_{r\in\mathcal{R}} \rho_{rs} y_r \geq 2k(S), \quad \forall S \in \mathcal{S}, \tag{17}$$

  where ρ_{rs} is the times route $r \in \mathcal{R}$ traverses an edge of $\delta(S)$.
- Such cuts do not change the pricing problem that remains "robust" [Fukasawa et al. 2006].
- Let v_S be the dual variable of (17), the pricing problem can be solved as before on the matrix $\overline{d}_{ij} = d_{ij} - \frac{1}{2}(u_i + u_j) - \sum_{S\in\mathcal{S}_{ij}} v_S$, where $\mathcal{S}_{ij} = \{S \in \mathcal{S} \,:\, \{i,j\} \in \delta(S)\}$, $\forall\{i,j\} \in E$.

General Description of Bounding Procedure H

- H computes 3 lower bounds, LB_1, LB_2 and LB_3 s.t. $LB_1 \leq LB_2 \leq LB_3$, corresponding to 3 dual solutions $(\mathbf{u^1}, \mathbf{v^1})$, $(\mathbf{u^2}, \mathbf{v^2})$, $(\mathbf{u^3}, \mathbf{v^3})$, of the linear relaxation of SP plus RCC (17).
- The master problem is solved with the dual ascent procedure, describe before, based on Theorem 1.
- LB_1 is obtained by using q-routes as columns.
- LB_2 is obtained by using ng-routes as columns.
- LB_3 is obtained by using elementary routes as columns.
- RCC (17) are separated heuristically once at the beginning and are cuts violated by the linear relaxation of $(2I)$.

Outline of Bounding Procedure H

1. Solve the linear relaxation of $(2I)$.
2. Separate a set $\mathcal{S}$ of violated RCC
3. Compute the dual solution $(\mathbf{u^1}, \mathbf{v^1})$, of problem $(SP) + RCC$, of cost LB_1 with a CG method, where:
   - Columns are q-routes and are generated by DP.
   - The master is solved with Theorem 1.
   - $\mathcal{S}$ is the set of rounded capacity constraints.
4. Compute the dual solution $(\mathbf{u^2}, \mathbf{v^2})$, of problem $(SP) + RCC$, of cost LB_2 with a CG method, where:
   - Columns are ng-routes and are generated by DP.
   - The master is solved with Theorem 1.
   - $\mathcal{S}$ is the set of rounded capacity constraints.
   - The master problem is initialized by using $(\mathbf{u^1}, \mathbf{v^1})$.
5. Compute the dual solution $(\mathbf{u^3}, \mathbf{v^3})$, of problem $(SP) + RCC$, of cost LB_3 with a CG method, where:
   - Columns are elementary routes and are generated by DP.
   - The master is solved with Theorem 1.
   - $\mathcal{S}$ is the set of rounded capacity constraints.
   - The master problem is initialized by using $(\mathbf{u^2}, \mathbf{v^2})$.

Adding Cuts from (SP)

- Lower bound LB_3 can be improved by adding cuts from the set packing/partitioning (e.g., clique inequalities).
- Such cuts make the pricing problem "non-robust", so the algorithms for solving the subproblem need relevant changes.
- A class of tractable, but still effective, cuts is the Subset-Row Inequalities (SRI) - introduced by [Jepsen et al. 2008]:
  - $\mathcal{C} \subseteq \{C \subseteq V : |C| = 3\}$
  - $\mathcal{R}(C) \subseteq \mathcal{R}$ routes that visit at least two of the customers in $C \in \mathcal{C}$

$$\sum_{r \in \mathcal{R}(C)} y_r \leq 1, \quad \forall C \in \mathcal{C}. \tag{18}$$

- SRI (18) can be separated by complete enumeration and can be handled in the pricing problem by properly tailoring dominance rules.
- Let $\mathbf{g}$ be the vector of dual variables associated with (18).

Multiple Feasible Dual Solutions

- Lower bound LB_3 can be also improved by using multiple feasible dual solutions to eliminate columns.
- Consider a generic IP problem with n variables and m constraints

$$z(F) = \min \mathbf{cx} \tag{19}$$
$$s.t.\ \mathbf{Ax} = \mathbf{b}, \tag{20}$$
$$\mathbf{x} \in \mathbb{B}^n. \tag{21}$$

- LF linear relaxation of F
- $z(LF)$ optimal solution cost of LF
- D dual of LF
- z_{UB} upper bound to $z(F)$

Multiple Feasible Dual Solutions (2)

- Let $\mathbf{w}'$ be a feasible D solution of cost z_{LB}.
- Any optimal F solution $\mathbf{x}^*$ satisfies $z(F) = z_{LB} + \sum_{j \in J} c'_j$, where c'_j is the reduced cost of x_j w.r.t. $\mathbf{w}'$ and $J = \{j : x^*_j = 1, j = 1, \ldots, n\}$.
- Then, any variable x_j s.t. $z_{LB} + c'_j > z_{UB}$ can be removed from (F) because cannot be in any optimal solution.
- The solution cost, $z(LF')$, of the linear relaxation of the resulting problem (F') is s.t. $z(LF') \geq z(LF)$.

Multiple Feasible Dual Solutions (3)

$$
\begin{array}{llllllll}
 & Min & x_1 & +x_2 & +x_3 & +4x_4 & +3x_5 & +3.5x_6 & \\
[w_1] & & x_1 & & & +x_4 & +x_5 & +x_6 & = 1 \\
[w_2] & & & +x_2 & & +x_4 & +x_5 & & = 1 \\
[w_3] & & & & x_3 & +x_4 & & +x_6 & = 1 \\
[w_4] & & x_1 & +x_2 & & & +x_5 & & = 1 \\
[w_5] & & x_1 & & +x_3 & & & +x_6 & = 1 \\
[w_6] & & & x_2 & +x_3 & & & & = 1 \\
 & & \multicolumn{6}{l}{x_i \in \{0,1\},\ i = 1,\dots,6}
\end{array}
$$

- $z(F) = 4$ and $z(LF) = 3.5$
- $z_{UB} = 4.5$ with $\mathbf{x} = (0,1,0,0,0,1)$
- $z_{LB} = 2$ with $\mathbf{w}' = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}) \Rightarrow z_{UB} - z_{LB} = 2.5$
- $\mathbf{c}'(\mathbf{w}') = (0,0,0,3,2,2.5) \Rightarrow$ remove $x_4 \Rightarrow z(LF') = 4$

Bounding Procedure *CCG*

- *CCG* is a column-and-cut generation algorithm that computes lower bound LB_4 corresponding to a dual solution $(\mathbf{u^4}, \mathbf{v^4}, \mathbf{g^4})$ of the linear relaxation of (SP) plus RCC (17) and SRI (18).
- *CCG* is executed after procedure H.
- The master problem is solved with the simplex.
- The pricing problem is solved with DP recursions.
- We use multiple feasible dual solutions, so each column of negative reduced cost w.r.t. the current dual solution that is generated is such that its reduced cost w.r.t. $(\mathbf{u^3}, \mathbf{v^3})$ is less than the gap between a known upper bound z_{UB} to the CVRP and LB_3.
- The set $\mathcal{S}$ of RCC is inherited from bounding procedure H.
- SRI inequalities are separated by complete enumeration.

Outline of the Exact Method

1. Call bounding procedure H to compute a feasible dual solution $(\mathbf{u^3}, \mathbf{v^3})$ of cost LB_3 of the linear relaxation of (SP) plus RCC.
2. Call bounding procedure CCG to compute a feasible dual solution $(\mathbf{u^4}, \mathbf{v^4}, \mathbf{g^4})$ of cost LB_4 of the linear relaxation of (SP) plus RCC and SRI.
3. Generate, via DP, the set $\hat{\mathcal{R}} \subseteq \mathcal{R}$ of routes s.t. $c_r^3 \leq z_{UB} - LB_3$ and $c_r^4 \leq z_{UB} - LB_4$, $\forall r \in \hat{\mathcal{R}}$, where c_r^3 and c_r^4 are the reduced costs of route r w.r.t. $(\mathbf{u^3}, \mathbf{v^3})$ and $(\mathbf{u^4}, \mathbf{v^4}, \mathbf{g^4})$, respectively.
4. Compute an optimal CVRP solution by solving, with an IP solver, problem (SP) by replacing the set of routes $\mathcal{R}$ with $\hat{\mathcal{R}}$.

If the DP recursion for generating routes runs out of memory in any of the first three steps, the algorithms terminates prematurely without providing any optimal solution.

Computational Results on the CVRP

- The exact method (hereafter *BMR*) was tested on 6 classes, *A*, *B*, *E*, *M*, *F*, *P*, of instances from the literature.
- All tests were performed on IBM Intel Xeon X7350@2.93 GHz [a].
- We compare the computational results achieved with the following exact methods:
  - [Lysgaard et al. 2004] (*LLE*) - Intel Celeron 700 MHz ([a] $\approx 10x$ faster)
  - [Fukasawa et al. 2006] (*FLL*) - Pentium 4 2.4 GHz ([a] $\approx 3x$ faster)
  - [Baldacci et al. 2008] (*BCM*) - Pentium 4 2.6 GHz ([a] $\approx 3x$ faster)

Computational Results on the CVRP (2)

		BMR			BCM			FLL					LLE		
Class	*NP*	*Opt*	*LB*	*CPU*	*Opt*	*LB*	*CPU*	*Opt*	*BCP*	*BC*	*LB*	*CPU*	*Opt*	*LB*	*CPU*
A	22	22	99.9	30	22	99.8	118	22	20	2	99.2	1,961	15	97.9	6,638
B	20	20	99.9	67	20	99.8	417	20	6	14	99.5	4,763	19	99.4	8,178
E-M	12	9	99.8	303	8	99.4	1,025	9	7	2	98.9	126,987	3	97.7	39,592
F	3	2	100.0	164				3	0	3	99.9	2,398	3	99.9	1,046
P	24	24	99.8	85	22	99.7	187	24	16	8	99.2	2,892	16	97.7	11,219
Avg			99.9	92		99.7	323				99.3	17,409		98.4	9,935
Tot	81	77			72			78	49	29			56		

BMR: our method - BCM: [Baldacci et al. 2008] - FLL: [Fukasawa et al. 2006] - LLE: [Lysgaard et al. 2004]

Outline

Variants considered

- The Heterogenous Vehicle Routing Problem (HVRP)
- The Pickup and Delivery Problem with Time Windows (PDPTW):
  - The Vehicle Routing Problem with Time Windows (VRPTW);
  - The Traveling Salesman Problem with Time Windows (TSPTW);
- The Periodic Vehicle Routing Problem (PVRP);
- The Capacitated Location-Routing Problem (LRP).
- The Two-Echelon Vehicle Routing Problem (2EVRP)

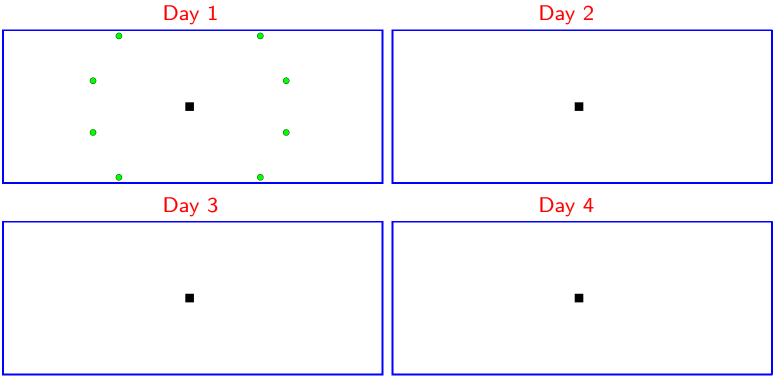
The Heterogenous Vehicle Routing Problem (HVRP)

Problem	Vehicle fixed costs	Vehicle dependent routing costs	Heterogenous vehicle fleet	Limited fleet
HVRP	Yes	Yes	Yes	Yes
CVRP	No	No	No	Yes
FSMF	Yes	No	Yes	No
FSMFD	Yes	Yes	Yes	No
HD / SDVRP	No	Yes	Yes	Yes
FSMD	No	Yes	Yes	No
MDVRP	No	Yes	No	No

- CVRP: capacitated vehicle routing problem.
- FSMF: fleet size and mix CVRP with fixed vehicle costs.
- FSMFD: fleet size and mix CVRP with fixed vehicle costs, vehicle dependent routing costs.
- HD: heterogenous CVRP with vehicle dependent routing costs.
- SDVRP: Site-Dependent CVRP.
- FSMD: fleet size and mix CVRP with vehicle dependent routing costs.
- MDVRP: multi-depot CVRP.

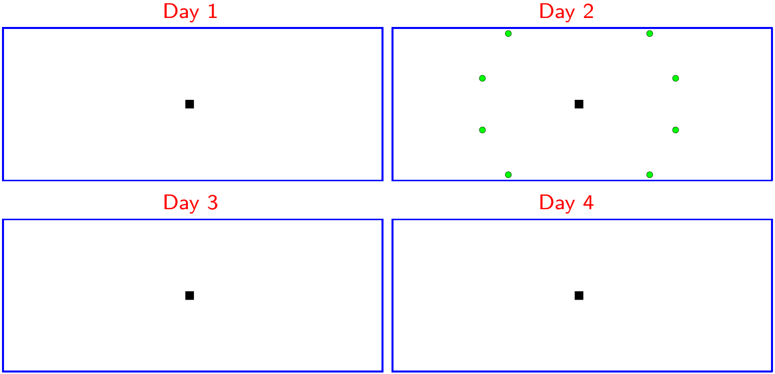
The Periodic Vehicle Routing Problem (PVRP)

4-day period: customers with *frequency* $= 1$



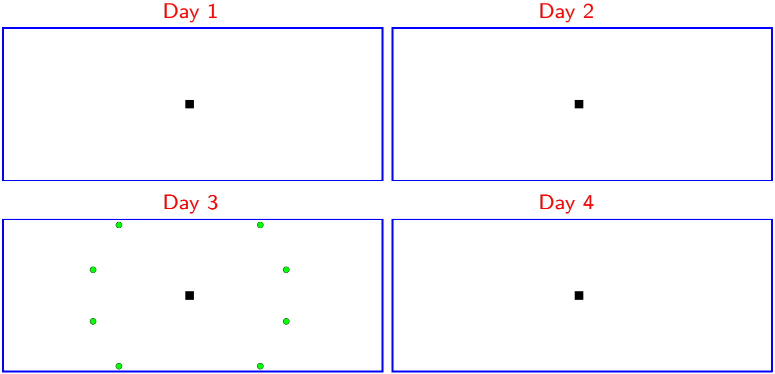
The Periodic Vehicle Routing Problem (PVRP) (2)

4-day period: customers with *frequency* $= 1$



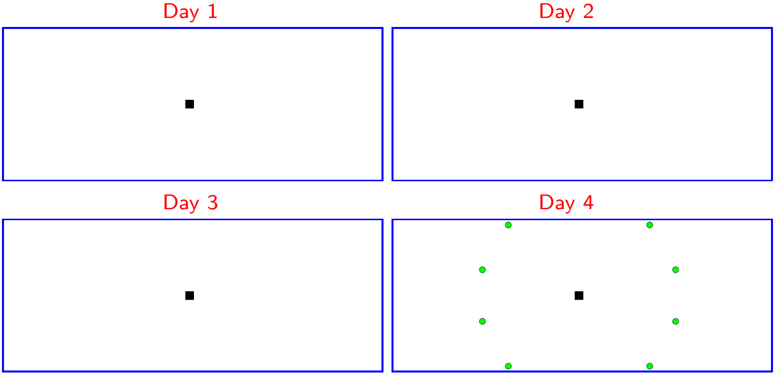
The Periodic Vehicle Routing Problem (PVRP) (3)

4-day period: customers with *frequency* $= 1$



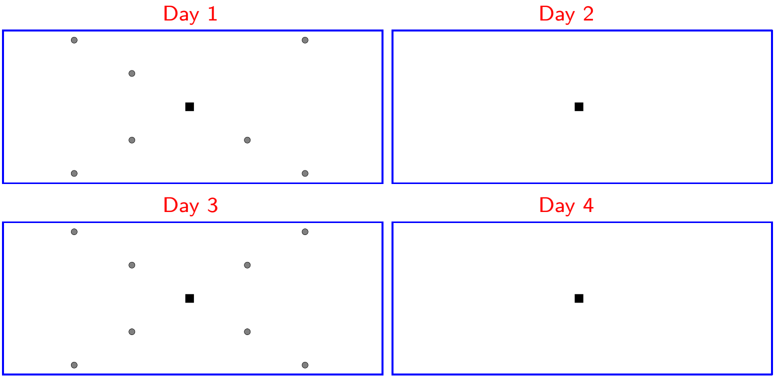
The Periodic Vehicle Routing Problem (PVRP) (4)

4-day period: customers with *frequency* $= 1$



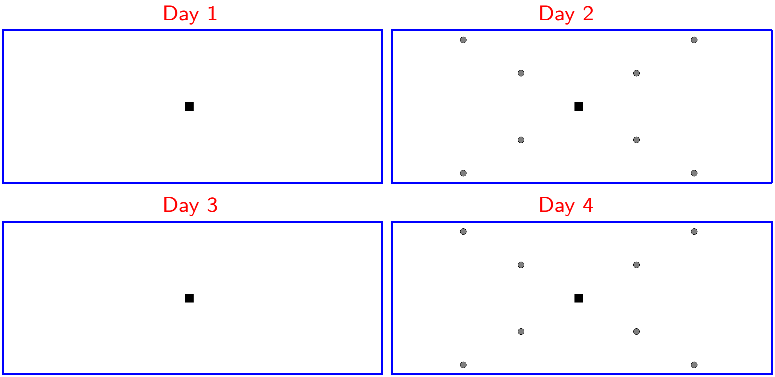
The Periodic Vehicle Routing Problem (PVRP) (5)

4-day period: customers with *frequency* $= 2$



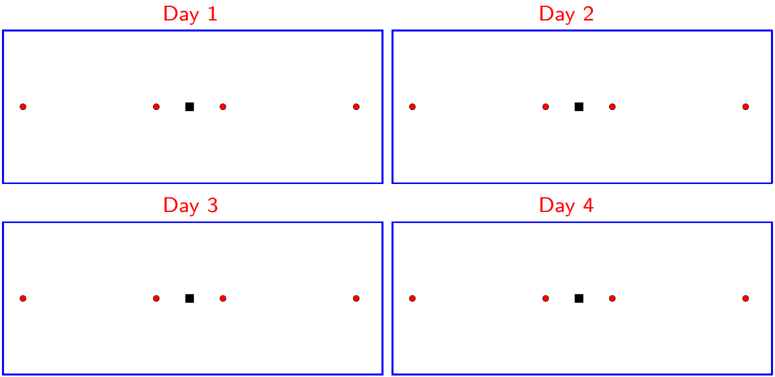
The Periodic Vehicle Routing Problem (PVRP) (6)

4-day period: customers with *frequency* $= 2$



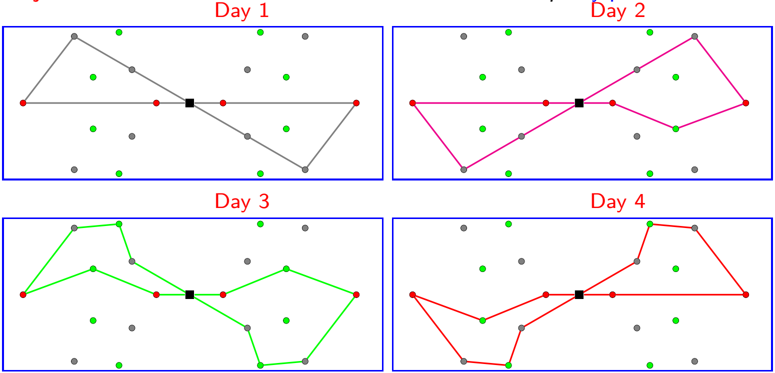
The Periodic Vehicle Routing Problem (PVRP) (7)

4-day period: customers with *frequency* $= 4$



The Periodic Vehicle Routing Problem (PVRP) (8)

- Problem: determine a day-combination for each customer and a set of at most m_k vehicle routes for each day k such that:
  - each route starts and finishes at the depot;
  - each route serves a total customer demand that is less than or equal to the vehicle capacity Q.
- Objective: minimize the total cost of the routes over the p-day period.



Summary of the computational results obtained over all variants

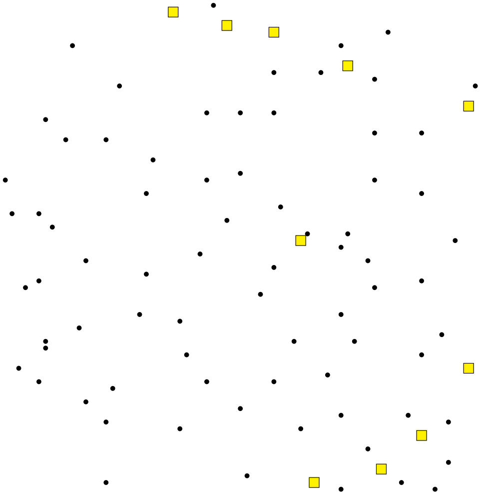
- Instances from the literature involving up to 199 customers and 15 depots

		Literature		CCG methods		
Variant	*#Inst*	*#OptLit*	*%OptLit*	*%LB*	*#Opt*	*%Opt*
CVRP	75	75	100.0	99.8	75	100.0
HVRP	12			99.6	10	83.3
FSMF	12	9	75.0	99.8	11	91.7
FSMFD	12	10	83.3	99.7	11	91.7
HD	8			99.2	7	87.5
FSMD	12	10	83.3	99.5	12	100.0
SDVRP	13			99.1	9	69.2
MDVRP	17			99.5	14	82.4
PDPTW	76	50	65.8	99.9	65	85.5
VRPTW	168	163	97.0	99.9	167	99.4
TSPTW	270	133	49.3	99.8	269	99.6
PVRP	68			99.2	41	60.3
	743	**450**	**74,1**		**691**	**92.3**

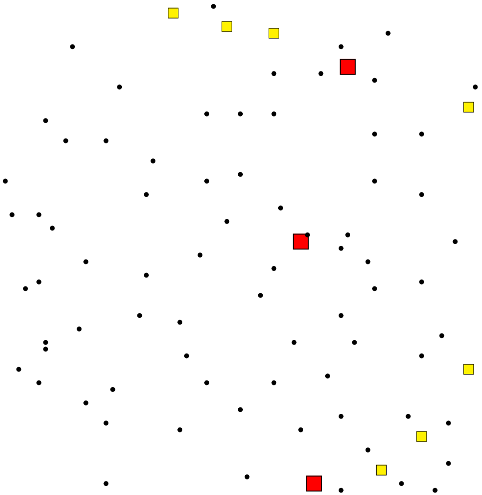
- *#OptLit*: total number of instances solved by the other methods

Outline

The Capacitated Location-Routing Problem (LRP)



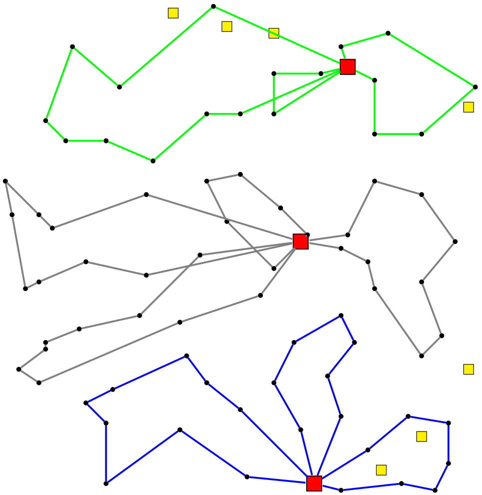
The Capacitated Location-Routing Problem (LRP) (2)



- LRP: opening a *set of depots*

  ...

The Capacitated Location-Routing Problem (LRP) (3)



- LRP: opening a *set of depots* and designing a set of routes so that:
  - (i) the total load of the routes operated from a depot does not exceed its capacity;
  - (ii) each customer is visited by exactly one route.
- Objective: minimize the sum of the fixed costs of the opened depots and the route costs.

Literature review

The LRP

- is $\mathcal{NP}$-hard.
- Surveys: [Laporte], [Laporte et al.], [Min et al.], [Nagy and Salhi].
- Heuristic methods: [Bruns and Klose], [Tuzun and Burke], [Wu et al.], [Albareda-Sambola et al.], [Prins et al.], [Prins et al.], [Prodhon et al.] and [Barreto et al.].
- Exact methods:
  - [Laporte et al.]: branch-and-cut algorithm for the LRP with uncapacitated depots;
  - [Akca et al.]: branch-and-price algorithm based on a set-partitioning formulation of the LRP;
  - [Belenguer et al.]: branch-and-cut algorithm based on a two-index formulation of the LRP.

Mathematical formulation - LRP

- $\mathscr{R}^k$: index set of all routes for depot $k \in Ns$.
- $\mathscr{R}_i^k \subset \mathscr{R}^k$: routes for depot $k \in Ns$ covering customer $i \in V_c$.
- c_ℓ^k, w_ℓ^k: cost and load of route $\ell \in \mathscr{R}^k$.
- R_ℓ^k: subset of customers visited by route $\ell \in \mathscr{R}^k$.
- x_ℓ^k: binary variable equal to 1 iff route $\ell \in \mathscr{R}^k$ is in solution.
- y_k: binary variable equal to 1 iff depot $k \in Ns$ is opened.

$$
\begin{aligned}
(F) \quad z(F) = \min & \sum_{k \in Ns} \sum_{\ell \in \mathscr{R}^k} c_\ell^k \xi_\ell^k + \sum_{k \in Ns} U_k y_k \\
s.t. \ & \sum_{k \in Ns} \sum_{\ell \in \mathscr{R}_i^k} \xi_\ell^k = 1, \quad \forall i \in V_c, \\
& \sum_{\ell \in \mathscr{R}^k} w_\ell^k \xi_\ell^k \le W_k y_k, \quad \forall k \in Ns, \\
& \xi_\ell^k \in \{0,1\}, \quad \forall \ell \in \mathscr{R}^k, \forall k \in Ns, \\
& y^k \in \{0,1\}, \quad \forall k \in Ns.
\end{aligned}
$$

Relaxation RF

- Let β_{ik} be the marginal routing cost for servicing customer $i \in V_c$ from depot $k \in Ns$ satisfying the following inequalities:

$$\sum_{i \in R_\ell^k} \beta_{ik} \leq c_\ell^k, \quad \forall \ell \in \mathscr{R}^k, \quad \forall k \in Ns. \qquad (MRCi)$$

- ξ_{ik}: binary variable equal to 1 iff $i \in V_c$ is supplied from depot $k \in Ns$.

The following integer problem RF provides a valid lower bound on the LRP:

$$\begin{aligned}
(RF) \quad z(RF) = \min & \sum_{k \in Ns} \sum_{i \in V_c} \beta_{ik} \xi_{ik} + \sum_{k \in Ns} U_k y_k \\
s.t. \; & \sum_{k \in Ns} \xi_{ik} = 1, \quad \forall i \in V_c, \\
& \sum_{i \in V_c} q_i \xi_{ik} \leq W_k y_k, \quad \forall k \in Ns, \\
& \xi_{ik} \in \{0,1\}, \quad \forall i \in V_c, \forall k \in Ns, \\
& y_k \in \{0,1\}, \quad \forall k \in Ns.
\end{aligned} \qquad (22)$$

Relaxation RF (2)

- Let LF be the LP-relaxation of F and let $z(LF)$ be its optimal solution cost.
- Lower bound $z(RF)$ achieved by RF can be greater than $z(LF)$.

Theorem

Let $z(RF(\beta))$ be the optimal solution of RF for a given solution β of (MRCi). The following relation holds:

$$\max_{\boldsymbol{\beta}}\{z(RF(\beta))\} \geq z(LF) \tag{23}$$
$$s.t. \quad (MRCi)$$

and inequality (23) can be strict.

Relaxation $\overline{RF}$ and bounding procedures DP^1 and DP^2

- We further relax RF replacing constraints (22) with the following constraints:

$$\sum_{k \in Ns} \sum_{i \in V_c} q_i \xi_{ik} = \sum_{i \in V_c} q_i. \tag{24}$$

- Denote by $\overline{RF}$ the resulting problem and by $z(\overline{RF})$ its optimal solution cost.
- Both procedures DP^1 and DP^2 are based on the following theorem:

Theorem

Let $\lambda_i \in \mathbb{R}$, $\forall i \in V_c$, be a set of penalties associated to the customers. A feasible solution β_{ik} of (MRCi) is given by setting:

$$\beta_{ik} = q_i \min_{\ell \in \mathscr{R}_i^k} \left\{ \frac{c_\ell^k - \lambda(R_\ell^k)}{w_\ell^k} \right\} + \lambda_i, \quad \forall i \in V_c,\ \forall k \in Ns,$$

where $\lambda(R_\ell^k) = \sum_{i \in R_\ell^k} \lambda_i$.

Relaxation $\overline{RF}$ and bounding procedures DP^1 and DP^2 (2)

- Bounding procedures DP^1 and DP^2 use different methods for computing values β_{ik}, but use the same dynamic programming algorithm for solving $\overline{RF}$.
  - DP^1 is based on q-route relaxation;
  - DP^2 uses column generation for computing β_{ik}.
- DP^2 is executed after DP^1 and uses the solution β^1_{ik} achieved by DP^1 to generate the initial master problem.
- We denote by β^2_{ik} the solution of (MRCi) achieved by DP^2 and by $LD1$ and $LD2$ the lower bounds obtained by DP^1 and DP^2, respectively.

Exact method

- Let $\mathscr{D} = \{D \subseteq Ns : \sum_{k \in D} W_k \geq \sum_{i \in V_c} q_i\}$.
- An optimal LRP solution can be computed as follows:

$$z(F) = \min_{D \in \mathscr{D}} \Big(\sum_{k \in D} U_k + z(F(D)) \Big)$$

where $z(F(D))$ is the optimal solution cost of the following problem:

$$\begin{aligned}
F(D) \quad z(F(D)) = \min & \sum_{k \in D} \sum_{\ell \in \mathscr{R}^k} c^k_\ell \xi^k_\ell \\
s.t. & \sum_{k \in D} \sum_{\ell \in \mathscr{R}^k_i} \xi^k_\ell = 1, \quad \forall i \in V_c, \\
& \sum_{\ell \in \mathscr{R}^k} w^k_\ell \xi^k_\ell \leq W_k, \quad \forall k \in D, \\
& \sum_{\ell \in \mathscr{R}^k} \xi^k_\ell \geq 1, \quad \forall k \in D, \\
& \xi^k_\ell \in \{0, 1\}, \quad \forall \ell \in \mathscr{R}^k, \forall k \in D.
\end{aligned}$$

- Let $\mathscr{R}(D) = \bigcup_{k \in D} \mathscr{R}^k$.

Exact method (2)

(A) Compute a lower bound on the LRP. Execute in sequence the bounding procedures DP^1 and DP^2.

(B) Generate the family of depot subsets $\mathscr{D}$. Let $LWB(D)$ be a valid lower bound on problem $F(D)$ computed as follows:

$$LWB(D) = \sum_{i \in V_c} \min_{k \in D} \{\beta^2_{ik}\}.$$

Let $U(D) = \sum_{k \in D} U_k$ and let $\mathscr{D}$ be the family of depot subsets such that:

$$\mathscr{D} = \{D \subseteq Ns : \sum_{k \in D} W_k \geq \sum_{i \in V_c} q_i \text{ and } U(D) + LWB(D) < z_{UB}\},$$

where z_{UB} is a known upper bound on the LRP.

Exact method (3)

(C) Solve the LRP.

(1) Initialize $z(F) = z_{UB}$, $LB = z_{UB}$ and $\overline{\mathscr{D}} = \emptyset$.
(2) If $\mathscr{D} = \emptyset$ then stop. Otherwise let $D \in \mathscr{D}$ be such that $U(D) + LWB(D) = \min_{D' \in \mathscr{D}}\{U(D') + LWB(D')\}$. Remove D from $\mathscr{D}$. If $U(D) + LWB(D) \geq z(F)$ then stop.
(3) Solve problem $F(D)$:

(i) Execute DP^1 and DP^2 on the reduced LRP problem obtained by replacing the depot set Ns with D. Let $LD2(D)$ be the lower bound obtained by DP^2. If $LD2(D) \geq z(F)$ return to Step 2, otherwise update $LB = \min\{LB, LD2(D)\}$.
(ii) Compute lower bound $LCG(D)$. If $U(D) + LCG(D) \geq z(F)$ return to Step 2. Otherwise update $LB = \min\{LB, U(D) + LCG(D)\}$.
(iii) Solve problem $F(D)$ to optimality. Update $z(F) = \min\{z(F), U(D) + z(F(D))\}$, $\overline{\mathscr{D}} = \overline{\mathscr{D}} \cup \{D\}$. Return to Step 2.

- $LCG(D)$ and the optimal solution of problems $F(D)$ are based on the exact algorithm for the Heterogeneous VRP of [Baldacci and Mingozzi 2008].

Computational results

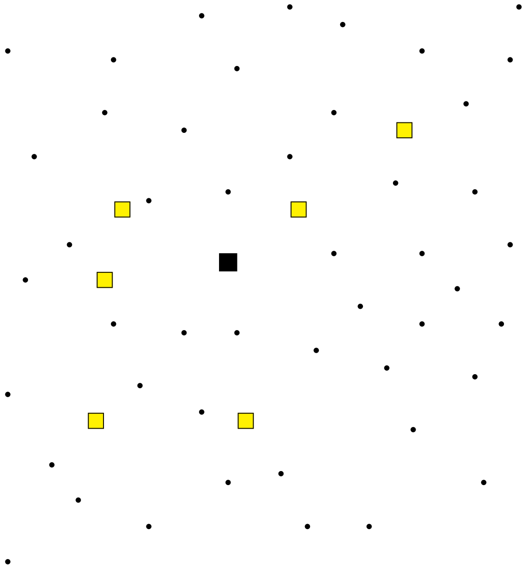
- We considered 55 LRP instances taken from the literature partitioned into 4 classes and involving up to 100 customers and 10 depots.
- Data and best known upper bounds are available at:
  - `http://www.isima.fr/lacomme/lrp/lrp.html`;
  - `http://sweet.ua.pt/~iscf143/_private/SergioBarretoHomePage.htm`.
- Computing times in seconds of:
  - Our method: Intel Xeon E5310 Workstation at 1.6 GHz with 8 GB RAM;
  - [Belenguer et al.]: Pentium 4 2.66 GHz with 2 GB of RAM;
  - [Akca et al.]: Linux-based workstation at 1.8 GHz with 2 GB RAM.

LRP: Summary of the computational results

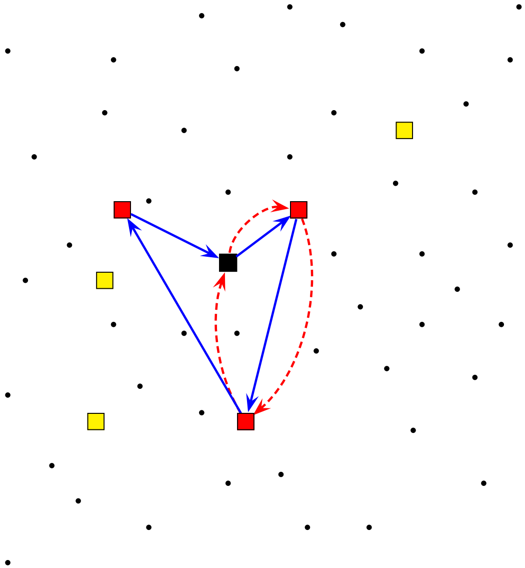
		Belenguer et al.			Akca et al.			Exact Method		
Class	*#inst*	*%LB*	t_{tot}	*#opt*	*%LB*	t_{tot}	*#opt*	*%LB*	t_{tot}	*#opt*
1	24	94.0	2.9	5				99.2	1,234.0	17
2	12	93.6	366.5	12	95.9	2,371.4	8	99.8	165.8	12
3	15	94.7	23.7	8	93.5	970.6	4	99.9	1,381.4	15
4	9							99.5	11,200.7	6
	60			**25**			**12**		**3,495.5**	**50**

- Our method: Intel Xeon E5310 Workstation at 1.6 GHz with 8 GB RAM;
- [Belenguer et al.]: Pentium 4 2.66 GHz with 2 GB of RAM;
- [Akca et al.]: Linux-based workstation at 1.8 GHz with 2 GB RAM.

The Two-Echelon Vehicle Routing Problem (2EVRP)

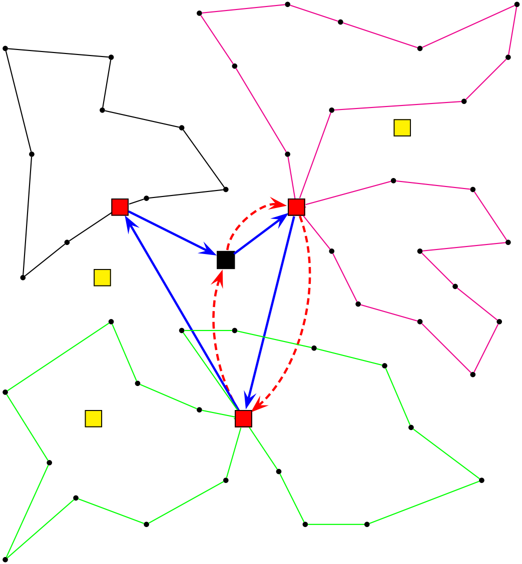


The Two-Echelon Vehicle Routing Problem (2EVRP) (2)



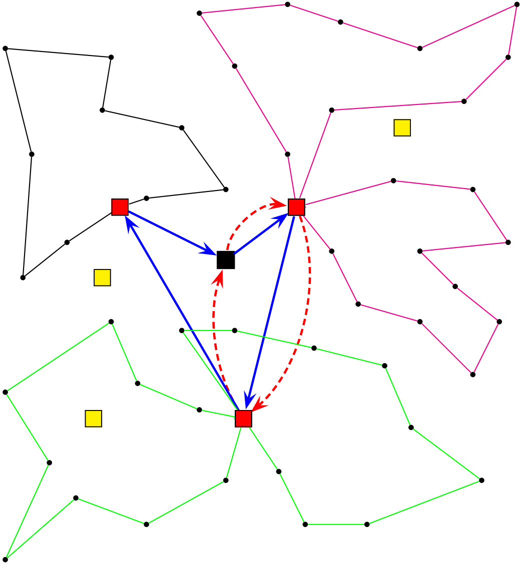
- To design a set of 1^{st}-level routes to supply the satellites and

The Two-Echelon Vehicle Routing Problem (2EVRP) (3)



- To design a set of 1^{st}-level routes to supply the satellites and
- To design a set of 2^{nd}-level routes from the satellites visited to supply the customers, so that
  - (i) the total load of the 2^{nd}-level routes operated from each satellite is equal to the quantity received from the 1^{st}-level route;
  - (ii) each customer is visited by exactly one route.

The Two-Echelon Vehicle Routing Problem (2EVRP) (4)



- To design a set of 1^{st}-level routes to supply the satellites and
- To design a set of 2^{nd}-level routes from the satellites visited to supply the customers, so that
  - (i) the total load of the 2^{nd}-level routes operated from each satellite is equal to the quantity received from the 1^{st}-level route;
  - (ii) each customer is visited by exactly one route.
- Objective: to minimize the sum of the two level routing costs.

Literature Review

The 2E-VRP:

- is $\mathcal{NP}$-hard.
- Heuristic methods:
  - J. Gonzales Feliu *et al.*, Technical Report DEIS, 2007;
  - T.G. Crainic *et al.*, Technical Report CIRRELT, 2008;
  - T.G. Crainic *et al.*, Technical Report CIRRELT, 2009;
  - T.G. Crainic *et al.*, Technical Report CIRRELT, 2010;
  - G. Perboli, R. Tadei and D. Vigo, Transportation Science, 2011, forthcoming.
- Exact methods (branch-and-cut):
  - "The Two-Echelon Capacitated Vehicle Routing Problem: models and math-based heuristics", G. Perboli, R. Tadei and D. Vigo, Transportation Science, 2011, forthcoming;
  - "A Branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle routing problem", M. Jepsen, S. Spoorendonk, S. Ropke, Submitted for publication, 2011.

Notation

- $\mathscr{M}$: index set of all 1^{st}-level routes.
- $\mathscr{R}_k$: index set of all 2^{nd}-level routes for depot $k \in Ns$.
- $\mathscr{M}_k$: index set of all 1^{st}-level routes passing through satellite k.
- $\mathscr{R}_{ki} \subseteq \mathscr{R}_k$: 2^{nd}-level routes from satellite $k \in Ns$ covering customer $i \in V_c$.
- R_ℓ^k: subset of customers visited by 2^{nd}-level route $\ell \in \mathscr{R}_k$.
- R_r: subset of satellites visited by 1^{st}-level route $r \in \mathscr{M}$.
- c_ℓ^k, w_ℓ^k: cost and load of 2^{nd}-level route $\ell \in \mathscr{R}_k$.
- g_r: the cost of a 1^{st}-level route $r \in \mathscr{M}$.

- x_ℓ^k: binary variable equal to 1 iff 2^{nd}-level route $\ell \in \mathscr{R}_k$ is in solution.
- y_r: binary variable equal to 1 iff 1^{st}-level route $r \in Ns$ is in solution.
- q_r^k: non negative variable representing the quantity delivered by 1^{st}-level route r to satellite $k \in R_r$.

Mathematical Formulation

$$
\begin{aligned}
(F) \quad z(F) = \min & \sum_{k \in Ns} \sum_{\ell \in \mathscr{R}_k} c_\ell^k x_\ell^k + \sum_{r \in \mathscr{M}} g_r y_r \\
s.t. \quad & \sum_{k \in Ns} \sum_{\ell \in \mathscr{R}_{ki}} x_\ell^k = 1, \quad \forall i \in V_c \\
& \sum_{k \in Ns} \sum_{\ell \in \mathscr{R}_k} x_\ell^k \leq m_2, \\
& \sum_{r \in \mathscr{M}} y_r \leq m_1, \\
& \sum_{r \in \mathscr{M}_k} q_r^k = \sum_{\ell \in \mathscr{R}_k} w_\ell^k x_\ell^k, \quad \forall k \in Ns \\
& \sum_{k \in R_r} q_r^k \leq Q_1 y_r, \quad \forall r \in \mathscr{M} \\
& x_\ell^k \in \{0,1\} \quad \forall \ell \in \mathscr{R}_k, \forall k \in Ns \\
& y_r \in \{0,1\} \quad \forall r \in \mathscr{M}, \text{and } q_r^k \geq 0 \quad \forall r \in \mathscr{M}, \forall k \in R_k.
\end{aligned}
$$

Relaxation RF

- Let β_{ik} be the Marginal Routing Cost (MRC) for servicing customer $i \in V_c$ from satellite $k \in Ns$ satisfying the following inequalities:

$$\sum_{i \in R_\ell^k} \beta_{ik} \leq c_\ell^k, \quad \forall \ell \in \mathscr{R}_k, \quad \forall k \in Ns. \quad (MRC)$$

- ξ_i: binary variable equal to 1 iff $i \in V_c$ is supplied from route $r \in \mathscr{M}$.

We compute a lower bound ϕ_{rw} on the cost of delivering a load w to customers from the subset of satellites R_r visited by the 1^{st}-level route: $r \in \mathscr{M}$

$$\begin{aligned}
\phi_{rw} = \min \sum_{i \in V_c} (\min_{k \in R_r} \{\beta_{ik}\}) \xi_i \\
s.t. \sum_{i \in V_c} q_i \xi_i = w, \\
\xi_i \in \{0,1\}, \quad \forall i \in V_c,
\end{aligned}$$

we compute ϕ_{rw}, $w = Q_1^{\min}, \ldots, Q_1,$, $r \in \mathscr{M}$, using Dynamic Programming.

Relaxation *RF* (2)

Given functions ϕ_{rw} the following integer problem *RF* provides a valid lower bound on the 2E-VRP:

$$(RF) \quad z(RF) = \min \sum_{r \in \mathscr{M}} \sum_{w=Q_1^{\min}}^{Q_1} (g_r + \phi_{rw}) \xi_{rw}$$

$$s.t. \sum_{r \in \mathscr{M}} \sum_{w=Q_1^{\min}}^{Q_1} w \xi_{rw} = \sum_{i \in Nc} q_i,$$

$$\sum_{w=Q_1^{\min}}^{Q_1} \xi_{rw} \leq 1 \quad \forall r \in \mathscr{M},$$

$$\xi_{rw} \in \{0, 1\}, \quad \forall r \in \mathscr{M}, \forall w = Q_1^{\min}, \ldots, Q_1,$$

Problem *RF* is a Multiple Choice Knapsack Problem which is solved by DP.

Computational results

- We considered 2E-VRP instances generated by Perboli et al. (2011), partitioned into 3 classes (Set 2, 3 and 4) and involving up to 51 customers and 5 satellites.
- Set 4 contains instances with upper bounds (UBs) on the maximum number of vehicles available per satellite. These constraints can be easily considered in our exact method.
- Data available at `http://people.brunel.ac.uk/~mastjjb/jeb/orlib/vrp2einfo.html`;
- We compare our method with the exact methods of:
  - PTV: G. Perboli, R. Tadei and D. Vigo, Transportation Science, 2011, forthcoming;
  - JSR: M. Jepsen, S. Spoorendonk, S. Ropke, Submitted for publication, 2011;

An exemple of the computational results

Table: Comparison with the exact methods PTV and JSR on Set 2 instances

		PTV		JSR				BMRW				
Name	$z(F)$	$\%UB$	$\%gap$	$\%UB$	$\%LB$	$\%gap$	t_{tot}	$\%UB$	$\%LB$	t_{LB}	$\%gap$	t_{tot}
E-n22-k4-s6-17	417.07	100.0	0.0	100.0	96.7	0.0	0.2	100.0	100.0	0.5	0.0	0.5
E-n22-k4-s8-14	384.96	106.0	0.0	100.0	98.0	0.0	1.0	100.0	100.0	0.7	0.0	0.7
E-n22-k4-s9-19	470.60	100.0	0.0	113.1	90.4	0.0	12.4	100.0	100.0	1.2	0.0	1.2
E-n22-k4-s10-14	371.50	117.3	0.0	100.0	96.8	0.0	1.2	100.0	100.0	0.5	0.0	0.5
E-n22-k4-s11-12	427.22	100.0	0.0	104.1	94.7	0.0	3.2	100.5	100.0	1.3	0.0	1.3
E-n22-k4-s12-16	392.78	108.4	0.0	100.0	95.9	0.0	2.0	100.0	100.0	1.0	0.0	1.1
E-n33-k4-s1-9	730.16	100.9	0.0	100.0	87.3	0.0	49.4	100.0	100.0	37.6	0.0	37.6
E-n33-k4-s2-13	714.63	103.0	1.5	100.0	89.4	0.0	34.2	100.0	100.0	34.9	0.0	34.9
E-n33-k4-s3-17	707.48	104.5	1.7	113.2	91.0	0.0	1,126.8	105.8	100.0	48.1	0.0	48.1
E-n33-k4-s4-5	778.74	104.9	1.5	100.0	87.6	0.0	54.9	100.9	100.0	72.5	0.0	72.5
E-n33-k4-s7-25	756.85	100.0	1.6	100.0	86.0	0.0	87.5	101.0	100.0	47.1	0.0	47.1
E-n33-k4-s14-22	779.05	100.0	1.6	105.9	88.0	0.0	2.4	100.0	100.0	31.7	0.0	31.7
E-n51-k5-s3-18	**597.49**	100.0	2.6	100.0	92.6	4.5	-	100.0	99.8	23.7	0.0	25.8
E-n51-k5-s5-47	530.76	102.3	1.8	102.4	97.0	0.0	13.3	101.6	99.8	25.9	0.0	27.5
E-n51-k5-s7-13	**554.81**	100.0	4.1	100.0	94.4	1.6	-	100.2	98.9	37.3	0.0	55.1
E-n51-k5-s12-20	581.64	100.4	3.7	104.2	94.2	0.0	213.6	100.5	99.3	27.1	0.0	44.3
E-n51-k5-s28-48	**538.22**	100.0	2.0	100.0	95.5	0.8	-	100.0	99.7	40.1	0.0	44.0
E-n51-k5-s33-38	552.28	104.7	0.7	100.0	95.8	0.0	2,114.0	100.0	100.0	13.6	0.0	13.6
E-n51-k5-s3-5-18-47	530.76	102.2	2.8	103.3	94.4	0.0	84.0	100.0	99.9	259.2	0.0	260.8
E-n51-k5-s7-13-33-38	531.92	107.5	3.6	102.7	94.6	0.0	3,642.8	100.0	99.4	263.6	0.0	266.6
E-n51-k5-s12-20-28-48	527.63	113.8	1.5	109.4	95.5	0.0	798.7	100.0	99.6	71.8	0.0	74.2
Avg./Solved		103.6	7	102.8	93.1	18	457.9	100.5	99.8		21	53.6

Bibliography

[Ascheuer et al. 2001] N. Ascheuer, M. Fischetti, and M. Grötschel. Solving the Asymmetric Traveling Salesman Problem with Time Windows by Branch-and-Cut. *Mathematical Programming*, 2001.

[Baker 1983] E. K. Baker. An Exact Algorithm for the Time-Constrained Traveling Salesman Problem. *Operations Research*, 1983.

[Balas and Simonetti 2001] E. Balas, and N. Simonetti. Linear Time Dynamic-Programming Algorithms for new Classes of Restricted TSPs: a Computational Study. *INFORMS Journal on Computing*, 2001.

[Baldacci et al. 2011a] R. Baldacci, E. Bartolini, A. Mingozzi, and A. Valletta. An Exact Algorithm for the Period Routing Problem. *Operations Research*, 2011a.

[Baldacci et al. 2011b] R. Baldacci, E. Bartolini, and A. Mingozzi. An Exact Algorithm for the Pickup and Delivery Problem with Time Windows. *Operations Research*, 2011b.

[Baldacci and Mingozzi 2008] R. Baldacci and A. Mingozzi. A Unified Exact Method for Solving Different Classes of Vehicle Routing Problems. *Mathematical Programming*, 2008.

[Baldacci et al. 2010] R. Baldacci, E. Bartolini, A. Mingozzi, and R. Roberti. An Exact Solution Framework for a Broad Class of Vehicle Routing Problems. *Computational Management Science*, 2010.

[Baldacci et al. 2006] R. Baldacci, L. Bodin, and A. Mingozzi. The multiple disposal facilities and multiple inventory locations rollon-rolloff vehicle routing problem. *Computers Oper. Res.*, 33(9):2667–2702, 2006.

[Baldacci et al. 2008] R. Baldacci, N. Christofides, and A. Mingozzi. An Exact Algorithm for the Vehicle Routing Problem based on the Set Partitioning Formulation with Additional Cuts. *Mathematical Programming*, 2008.

Bibliography (2)

[Baldacci et al. 2004] R. Baldacci, E. Hadjiconstantinou, and A. Mingozzi. An Exact Algorithm for the Capacitated Vehicle Routing Problem based on a Two-Commodity Network Flow Formulation. *Operations Research*, 2004.

[Baldacci et al. 2011c] R. Baldacci, A. Mingozzi, and R. Roberti. New Route Relaxation and Pricing Strategies for the Vehicle Routing Problem. *Operations Research*, forthcoming, 2011c.

[Baldacci et al. 2011d] R. Baldacci, A. Mingozzi, and R. Roberti. New State-Space Relaxations for Solving the Traveling Salesman Problem with Time Windows. *INFORMS Journal on Computing*, 2011d.

[Baldacci et al. 2011e] R. Baldacci, A. Mingozzi, R. Roberti, and R. Wolfler Calvo. An Exact Method for the 2-Echelon Capacitated Vehicle Routing Problem. *working paper*, 2011e.

[Baldacci et al. 2011f] R. Baldacci, A. Mingozzi, and R. Wolfler Calvo. An Exact Method for the Capacitated Location-Routing Problem. *Operations Research*, forthcoming, 2011f.

[Chabrier 2006] A. Chabrier. Vehicle Routing Problem with Elementary Shortest Path based Column Generation. *Computers & Operations Research*, 2006.

[Christofides et al. 1981] N. Christofides, A. Mingozzi, and P. Toth. State-space relaxation procedures for the computation of bounds to routing problems. *Networks*, 1981.

[Danna and Le Pape 2005] E. Danna and C. Le Pape. Accelerating Branch-and-Price with Local Search: A Case Study on the Vehicle Routing Problem with Time Windows. *Column Generation* Book, 2005.

[Dash et al. 2010] S. Dash, O. Günlük, A. Lodi, and A. Tramontani. A Time Bucket Formulation for the TSP with Time Windows. *INFORMS Journal on Computing*, 2010.

Bibliography (3)

[Desaulniers et al. 2008] G. Desaulniers, F. Lessard, and A. Hadjar. Tabu Search, Partial Elementarity, and Generalized k-Path Inequalities for the Vehicle Routing Problem with Time Windows. *Transportation Science*, 2008.

[Dumas et al. 1995] Y. Dumas, J. Desrosiers, E. Gélinas, M. M. Solomon. An Optimal Algorithm for the Traveling Salesman Problem with Time Windows. *Operations Research*, 1995.

[Feillet et al. 2004] D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen. An Exact Algorithm for the Elementary Shortest Path Problem with Resource Constraints: Application to some Vehicle Routing Problems. *Networks*, 2004.

[Focacci et al. 2002] F. Focacci, A. Lodi, and M. Milano. A Hybrid Exact Algorithm for the TSPTW. *INFORMS Journal on Computing*, 2002.

[Fukasawa et al. 2006] R. Fukasawa, H. Longo, J. Lysgaard, M. Poggi, M. Reis, E. Uchoa, and R. F. Werneck. Robust Branch-and-Cut-and-Price for the Capacitated Vehicle Routing Problem. *Mathematical Programming*, 2006.

[Irnich and Villeneuve 2006] S. Irnich and D. Villeneuve. The Shortest-Path Problem with Resource Constraints and k-Cycle Elimination for $k \geq 3$. *INFORMS Journal on Computing*, 2006.

[Jepsen et al. 2008] M. Jepsen, B. Petersen, S. Spoorendonk, and D. Pisinger. Subset-Row Inequalities Applied to the Vehicle-Routing Problem with Time Windows. *Operation Research*, 2008.

[Langevin et al. 1993] A. Langevin, M. Desrochers, J. Desrosiers, S. Gélinas, F. Soumis. A Two-Commodity Flow Formulation for the Traveling Salesman and the Makespan Problems with Time Windows. *Networks*, 1993.

[Li 2009] J. Q. Li. A Computational Study of Bi-directional Dynamic Programming for the Traveling Salesman Problem with Time Windows. *submitted*, 2009.

Bibliography (4)

[Lysgaard et al. 2004] J. Lysgaard, A. N. Letchford, and R. W. Eglese. A New Branch-and-Cut Algorithm for the Capacitated Vehicle Routing Problem. *Mathematical Programming*, 2004.

[Mingozzi et al. 1997] A. Mingozzi, L. Bianco, and S. Ricciardelli. Dynamic Programming Strategies for the Traveling Salesman Problem with Time Windows and Precedence Constraints. *Operations Research*, 1997.

[Mingozzi et al. 2011] A. Mingozzi, R. Roberti, and P. Toth. An Exact Method for the Multi-Trip Vehicle Routing Problem. *submitted*, 2011.

[Cornuéjols and Harche] G. Cornuéjols and F. Harche. Polyhedral study of the capacitated vehicle routing. *Math. Program.*, 60:21–52, 1993.

[Augerat et al.] P. Augerat, J.M. Belenguer, E. Benavent, A. Corberan, D. Naddef, and G. Rinaldi. Computational results with a branch and cut code for the capacitated vehicle routing problem. Technical Report 1 RR949-M, ARTEMIS-IMAG, Grenoble France, 1995.

[Gouveia] L. Gouveia. A result on projection for the vehicle routing problem. *J. Opl. Res.*, 85:610–624, 1995.

[Letchford and Salazar] A.N. Letchford and J.J. Salazar González. Projection results for vehicle routing. *Math. Program.*, 105(2-3):251–274, 2006.

[Chvátal] V. Chvátal. Edmonds polytopes and weakly hamiltonian graphs. *Math. Program.*, 5:29–40, 1973.

[Grötschel and Padberg] M. Grötschel and M.W. Padberg. On the symmetric traveling salesman problem: I and II. *Math. Program.*, 16:265–280, 1979.

Bibliography (5)

[Grötschel and M.W. Padberg] M. Grötschel and M.W. Padberg. Polyhedral theory. In E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, editors, *The traveling salesman problem: a guided tour of combinatorial optimization*, pages 231–305. John Wiley & Sons Ltd., Chichester, 1985.

[Akca et al.] Akca, Z., R. T. Berger, and T. K. Ralphs, 2009: A branch-and-price algorithm for combined location and routing problems under capacity restrictions. In *Operations Research and Cyber-Infrastructure*, Chinneck, J. W., Kristjansson, B., and Saltzman, M. J., editors, volume 47. Springer US, 309–330.

[Albareda-Sambola et al.] Albareda-Sambola, M., J. A. Diaz, and E. Fernandez, 2005: A compact model and tight bounds for a combined location-routing problem. *Computers & Operations Research*, **32**(3), 407–428.

[Barreto et al.] Barreto, S., C. Ferreira, J. Paixao, and B. S. Santos, 2007: Using clustering analysis location-routing in a capacitated problem. *European Journal of Operational Research*, **179**(3), 968–977.

[Belenguer et al.] Belenguer, J. M., E. Benavent, C. Prins, C. Prodhon, and R. Wolfler-Calvo,2009: A branch and cut algorithm for the capacitated location routing problem with depot and vehicle capacities.

[Bruns and Klose] Bruns, A. and A. Klose, 1996: A locate first - route second heuristic for a combined location-routeing problem. In *Operations Research Proceedings*, Zimmermann, U., Derigs, U., Gaul, W., Mohring, R., and Schuster, K., editors, Springer-Verlag, 49-54.

[Laporte] Laporte, G., 1988: Location-routing problems. In *Vehicle Routing: Methods and Studies*, Golden, B. and Assad, A., editors, North Holland, Amsterdam, 163-196.

[Laporte et al.] Laporte, G., F. Louveaux, and H. Mercure, 1989: Models and exact solutions for a class of stochastic location-routing problems. *European Journal of Operational Research*, **39**, 71–78.

Bibliography (6)

[Laporte et al.] Laporte, G., Y. Norbert, and D. Arpin, 1986: An exact algorithm for solving a capacitated location-routing problem. *Annals of Operations Research*, **6**, 293–310.

[Min et al.] Min, H., V. Jayaraman, and R. Srivastava, 1998: Combined location-routing problems: A synthesis and future research directions. *European Journal of Operational Research*, **108**, 1–15.

[Nagy and Salhi] Nagy, G. and S. Salhi, 2007: Location-routing: Issues, models and methods. *European Journal of Operational Research*, **177**(2), 649–672.

[Prodhon et al.] Prins, C. and C. Prodhon, A. Ruiz, P. Soriano and R. Wolfler Calvo, 2007: Solving the capacitated location-routing problem by a cooperative lagrangean relaxation-granular tabu search heuristic. *Transportation Science*, **41**(4), 470–483.

[Prins et al.] Prins, C., C. Prodhon, and R. Wolfler Calvo, 2006: A memetic algorithm with population management (ma vertical bar pm) for the capacitated location-routing problem. *Evolutionary Computation In Combinatorial Optimization, Proceedings*, **3906**, 183–194.

[Prins et al.] Prins, C., C. Prodhon, and R. Wolfler Calvo, 2006: Solving the capacitated location-routing problem by a grasp complemented by a learning process and a path relinking. *4OR*, **4**(3), 221–238.

[Tuzun and Burke] Tuzun, D. and L. Burke, 1999: A two-phase tabu search approach to the location routing problem. *European Journal of Operational Research*, **116**, 87–99.

[Wu et al.] Wu, T. H., C. Low, and J. W. Bai, 2002: Heuristic solutions to multi-depot location-routing problems. *Computers & Operations Research*, **29**(10), 1393–1415.

A non-elementary route

Return