ORSUM 2004

Seminar XI

Variable aggregation and dis-aggregation: As time goes by

Vicky Mak *Deakin University*

In the beginning of the world, there is linear program.

Problem 1

$$z = \max \sum_{j=1}^{n} c_j x_j$$

$$\text{s.t.} \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \; i = 1, \ldots, m$$

$$0 \leq x_j \leq r_i, \qquad j = 1, \ldots, n$$

An example:

$$z = \max 3x_{11} + 6x_{12} + 4x_{13} + 8x_{21} + 7x_{22} + 3x_{23}$$

$$\text{s.t.} \quad 6x_{11} + x_{12} + 3x_{13} + 4x_{21} + x_{22} + x_{23} \leq 50,$$

$$3x_{11} + 2x_{12} + 4x_{13} + 5x_{21} + 2x_{22} + 5x_{23} \leq 72,$$

$$x \in \mathbb{R}^n_+.$$

Now, imagine it's big (too many columns), and it's hard to solve.

Overview

$\longrightarrow$ Solve a smaller problem. (But how?)

$\longrightarrow$ Aggregate the columns (or the constraints if too many rows) (see Balas 1965 for an application in Transportation Problem).

$\longrightarrow$ How do we get a solution to the original problem?

Answer: Disaggregation

The topic for today: Column aggregation and disaggregation for Linear Program (LP).

Then, for Integer Program (IP).

Then, the applications.

Then, what to do next.

How does it work?

Let $\sigma = \{S_k \colon k = 1, \ldots, K\}$ be a partition of $\{1, \ldots, n\}$, so

$$\bigcup_{k=1}^{K} S_k = \{1, \ldots, n\}, \qquad \text{and}$$

$$S_k \cap S_j = \emptyset \quad \text{for all distinct } k, j.$$

Example: $S_1 = \{11, 12, 13\}$ and $S_2 = \{21, 22, 23\}$.

Then, assign weights. Let:

$$w^k \geq 0, \sum_{j \in S_k} w_j^k = 1, \forall k = 1, \ldots, K.$$

Example: $w^1 = (\frac{1}{2}, \frac{1}{4}, \frac{1}{4})$ and $w^2 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$.

For each $k \in \{1, \ldots, K\}$
Variables x_j for $j \in S_k$ is aggregated into one single variable y_k,

Example:
$\{x_{11}, x_{12}, x_{13}\} \to y_1$
$\{x_{21}, x_{22}, x_{23}\} \to y_2$

New cost: $\bar{c}^k = \sum_{j \in S_k} c_j w_j^k$

Example:
$\bar{c}^1 = (3, 6, 4)(\frac{1}{2}, \frac{1}{4}, \frac{1}{4})^T = 4$
$\bar{c}^2 = (8, 7, 3)(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T = 6$

New coefficient matrix:

Let $\widetilde{A}^k$ be the $m \times |S_k|$-submatrix of A.

Define: $\bar{A} = (\widetilde{A}^1 w^1, \ldots, \widetilde{A}^K w^K)$.

Example:

$$\widetilde{A}^1 = \begin{bmatrix} 6 & 1 & 3 \\ 3 & 2 & 4 \end{bmatrix} \qquad \widetilde{A}^2 = \begin{bmatrix} 4 & 1 & 1 \\ 5 & 2 & 5 \end{bmatrix}$$

$$\bar{A} = \left[\begin{bmatrix} 6 & 1 & 3 \\ 3 & 2 & 4 \end{bmatrix} \cdot (\tfrac{1}{2}, \tfrac{1}{4}, \tfrac{1}{4})^T, \begin{bmatrix} 4 & 1 & 1 \\ 5 & 2 & 5 \end{bmatrix} \cdot (\tfrac{1}{3}, \tfrac{1}{3}, \tfrac{1}{3})^T \right]$$

$$\bar{A} = \begin{bmatrix} 4 & 2 \\ 3 & 4 \end{bmatrix}$$

The aggregated problem

Problem 2

$$\widetilde{z} = \max \sum_k \bar{c}_k y_k$$

$$\text{s.t.} \quad \sum_k \bar{a}_{ik} y_k \leq b_i, \quad i = 1, \ldots, m$$

$$0 \leq y_k \leq \min_{j \in S_k} \left\{ \frac{r_j}{w_j^k} \right\}, \; k = 1, \ldots, K.$$

Example:

$$\widetilde{z} = \max 4y_1 + 6y_2$$

$$\text{s.t.} \quad 4y_1 + 2y_2 \leq 50,$$

$$3y_1 + 4y_2 \leq 72,$$

$$y_1, y_2 \geq 0.$$

One of the pioneers on error bounds: Zipkin, 1980

Zipkin: Result 1

If y is feasible for the aggregated problem, then the *fixed-weight disaggregation* $x^k = w^k y_k$ for $k = 1, \ldots, K$ is feasible for the original problem, and $cx = \bar{c}y$.

Consequently, if y^* is optimal for the aggregated problem, and $\tilde{x}^k = w^k y_k^*$ for $k = 1, \ldots, K$, then $c\tilde{x} = \bar{c}y^* = \bar{z} \leq cx^* = z$.

Okay, by solving the aggregation, and with fixed-weight disaggregation, we always get a lower bound!

Zipkin: Result 2

For any partition σ, there exists an optimal weight w^* such that $cx^* = \bar{c}y^*$, for x^*, y^* optimal to their respective problems.

$$w^{k*} = \begin{cases} \alpha, & \text{if} \quad y_k^* = 0, \\ \frac{x^{k*}}{y_k^*}, & \text{if} \quad y_k^* > 0, \end{cases}$$

for any arbitary α.

Zipkin: You can find w^* if you knew what x^* is!

Me: Yeah, and if you knew whether chicken (w^{k*}) came first or egg (y_k^*) came first.

Zipkin: Result 3 - Error bounds - this is the cool bit

Two error bounds:

(1) A priori bounds, without solving either of the problems; and

(2) A posteriori bounds, after solving the aggregated problem.

- The a priori bound is obtained by making use of constraint coefficients, cost coefficients, and some bounds on the optimal values of the variables for the original problem.
- The a posteriori bound needs the optimal dual values from the aggregated problem.

Zipkin: Result 3 - Error bounds - this is the cool bit

Let $\sigma' = \{S'_k : k = 1, \ldots, K'\}$ be any partition of N, and

$$\sum_{j \in S'_k} d_j x^*_j \leq p_k, \text{ for } k = 1 \ldots, K'$$

be bounds on x^* (which can be deduced from the constraints).

An example: (from Slide 2, if $\sigma' = \sigma$).

$$\begin{aligned} x_{11} + x_{12} + x_{13} &\leq 36, \\ x_{21} + x_{22} + x_{23} &\leq 36. \end{aligned}$$

Zipkin: Result 3 - Error bounds - this is the cool bit

The *A priori* bound

$$\epsilon_b = \sum_k \left[\max_{i,j} \left\{ \frac{c_j - \frac{\bar{c}_k a_{ij}}{\bar{a}_{ik}}}{d_j} : j \in S'_k,\ \bar{a}_{ik} > 0 \right\} \right]^+ \cdot p_k.$$

The *A posteriori* bound

$$\epsilon_a = \sum_k \left[\max_{j \in S'_k} \left\{ \frac{c_j - \mu A^j}{d_j} \right\} \right]^+ \cdot p_k.$$

a toy example

$$
\begin{aligned}
z = \max\ & 2.5x_1 + 3x_2 + 4x_3 + 5x_4 \\
\text{s.t. }\ & 4x_1 + 6x_2 + 7x_3 + 10x_4 \le 54, \\
& x_1 + 2x_2 + x_3 + 2x_4 \le 10, \\
& x_1, x_2, x_3, x_4 \ge 0.
\end{aligned}
$$

Let $S_1 = \{1, 2\}, S_2 = \{3, 4\}$, and $w^1 = w^2 = (\frac{1}{2}, \frac{1}{2})$:

$$
\begin{aligned}
\widetilde{z} = \max\ & 2.75y_1 + 4.5y_2 \\
\text{s.t. }\ & 4.5y_1 + 8.5y_2 \le 54, \\
& 1.5y_1 + 1.5y_2 \le 10, \\
& y_1, y_2 \ge 0.
\end{aligned}
$$

$z^* = 32$, and $\widetilde{z} = 28\frac{5}{6}$. $z^* - \widetilde{z} = 3\frac{1}{6}$, a priori bound $= 14\frac{2}{3}$, and a posteriori bound $= 5\frac{5}{8}$. So $28\frac{5}{6} \le z^* \le 28\frac{5}{6} + 5\frac{5}{8}$.

Zipkin: an algorithm for solving the original problem or at least obtaining better bounds

Problem 3

$$\tilde{z} = \max \sum_k \bar{c}_k y_k + \sum_j c_j x_j$$

$$\text{s.t.} \sum_k \bar{a}_{ik} y_k + \sum_j a_{ij} x_j \leq b_i, \quad i = 1, \ldots, m$$

$$0 \leq y_k \leq \min_{j \in S_k} \left\{ \frac{r_j}{w_j^k} \right\}, \qquad k = 1, \ldots, K$$

$$0 \leq x_j \leq r_j, \qquad j = 1, \ldots, n.$$

Starting with variables in the aggregated model only, iteratively generating “most valuable” variables to the original problem by finding best reduced cost column, (Ah ha! sounds familiar?)

Zipkin: an extension to algorithm

The scheme: Similar to before, but iteratively updating the weights according to:

$$\text{new}(w^k) = \begin{cases} \text{old } (w^k) & \text{if currently } \quad y_k + \sum_{j \in S_k} x_j = 0, \\ \frac{y_k w^k + x_k}{y_k + \sum_{j \in S_k} x_j} & \text{o.w.} \end{cases}$$

Thinking exercise: How would these two compare with just column generation? Hmm... these ones (may) have better bounds to start with: comparing with normal column generation when we don't have a decent strategy for choosing the initial set of variables.

Who's up to do some numerical comparisons?

A good review paper Rogers *et al.*, 1991

- Not just column aggregation and disaggregation, others as well.
- Loads of references in assorted topics: error bounds, theories, and applications.
- Iterative aggregation/disaggregation (IAD)
  - Miranker and Pan, 1980; Chatelin and Miranker, 1982. For solving systems of linear equations. (Method similar to the projection method of Galerkin approximations).
  - Mendelssohn, 1982. For LPs from infinite horizon MDPs.

Alternative approaches - Jörnsten and Leisten, 1992

Optimal Disaggregation: one problem for each of $k = 1, \ldots, K$.

Problem 4

$$\tilde{z} = \max \sum_{j \in S_k} c_j x_j$$

$$\text{s.t.} \sum_{j \in S_k} a_{ij} x_j = \bar{a}_{ik} y_k^*, \quad \forall i, k,$$

$$0 \leq x_j \leq r_j, \qquad j \in S_k.$$

The optimal disaggregation is at least as good as the fixed weight disaggregation. Hence, better bounds.

Their contributions: Proposed to solve the Aggregated Problem and Optimal Disaggregation iteratively. Experimented with 3 different schemes for updateing w^k. Numerically a subgradient optimization-type scheme is best.

Other papers in error bounds and applications

- Mendelssohn (1980) developed new error bounds by optimising the dual problems of the aggregated problem using a different approach, and implemented the algorithm on LPs generated from infinite-stage Markov Decision Processes (1982).
- Balas (1965) proposed row and column aggregation/dis-aggregation for the Transportation Problem.
- Alavrez, Chacon, Litvinchev, and Rangel (2001) worked on improving the error bounds on the Generalized Transportation Problem.
- The series of papers by Storoy *et al.*

A step further: Integer Programs - Hallefjord and Storoy, 1990

An old friend: Binary Integer Program.

$$z_{IP} = \max cx$$
$$\text{s.t. } Ax \leq b,$$
$$x \in \{0,1\}^n.$$

and the aggregation:

$$\tilde{z}_{IP} = \max \bar{c}y$$
$$\text{s.t. } \bar{A}y \leq b,$$
$$0 \leq y_k \leq \min_{j \in S_k} \left\{ \frac{1}{w_j^k} \right\},$$
$$y \in Z^K.$$

Results from Zipkin that still holds

1. If LP relaxation of the aggregated problem is solved, then the *a posteriori* bound still holds. (Of course!)

2. Optimal weights *does* exists, again, provided if we know x^* and knew the answer to that question!

Results from Zipkin that may not hold

1. Very often, fixed-weight dis-aggregation does *not* work.

2. No knowledge about Zipkin's Result 1 and its consequence. (But this is leaving us lots of rooms for research!)

Thinking exercise: With equality constraints, is disaggregation always feasible?

Consider

$$x_1 + 2x_2 + 3x_3 + 3x_4 + x_5 = 8,$$

with all x binary. If $S_1 = \{1, 2, 3\}$ and $S_2 = \{4, 5\}$. We have the aggregated constraint:

$$y_1 + y_2 = 8.$$

What if $y_1^* = 6$ and $y_2^* = 2$?

Under what conditions is disaggregation feasible?

Essence of Hallefjord and Storoy's paper

1. Two error bounds that are at least as good as Zipkin's.

$$\epsilon^1_{HS} = \min_{\bar{u}} \sum_{k=1}^{K'} \left[\max_{j \in S'_k} \left\{ (c_j - \bar{\mu}^1 A^j - \bar{\mu}^2)/d_j \right\} \right]^+ \cdot p_k,$$

for $\bar{\mu}^1$ and $\bar{\mu}^2$ optimal dual variables (often multiple of them really?) for the LP relaxation of the aggregated problems; so we solve:

$$\begin{aligned} \epsilon^1_{HS} &= \min \sum_{k=1}^{K} w_k \\ \text{s.t.} \quad & w_k \geq c_j - \bar{\mu}^1 A^j - \bar{\mu}^2_j, \forall j \in S'_k, \forall k; \text{ and} \end{aligned}$$

$$\epsilon^2_{HS} = \sum_{k=1}^{K'} \sum_{j=1}^{|S_k|} (c_j - \bar{\mu}^1 A^j - \bar{\mu}^2)/d_j.$$

Me: Why ϵ^2_{HS}? And, hope the number of rows is small :-)

Essence of Hallefjord and Storoy's paper

2. Arrange reduced cost of the original variables (using optimal dual variables from the aggregated problem) in descending order, add variables to generate minimal-cover inequalities until saturated. Benefit: may find better bounds on the run. Example: Solving AP of

$$
\begin{aligned}
5x_{11} + 6x_{12} + 9x_{13} + 14x_{21} + 15x_{22} + 16x_{23} &\leq 19, \\
5x_{11} + 5x_{12} + 7x_{13} + 11x_{21} + 12x_{22} + 13x_{23} &\leq 13; \text{ gets} \\
x_{11} + x_{21} + x_{22} + x_{23} &\leq 1 \text{ (cover).}
\end{aligned}
$$

(N.B. Cover cuts generated may acutally be satisifed by the current aggregated solution).

Question: The heuristic doesn't guarantee finding sufficient cuts! (Worked for the toy problem, what about other problems?)

Hallefjord and Storoy

Storoy went on researching aggregation/disggregation in LP:

1. S.Storoy: "Weights improvement in column aggregation", EJOR., Vol 73, 1994.
2. S.Storoy: "Optimal weights and degeneracy in aggregated linear programs" Operations Research Letters, Vol. 19, 1996.
3. K.Jörnsten, R.Leisten and S.Storoy: "Convergence aspects of adaptive clustering in variable aggregation", Computers & Oper. Res., Vol. 26, 1999.

Hallefjord had some ideas for IP, but unfortunately he passed away in 1996.

Litvinchev and Rangel, 1999

Translated their results for non-linear programming to linear programming. More bounds, and for problems proposed in Hallefjord and Storoy, managed to generate minimal-cover inequalities that are actually violated by the disaggregated solutions.

Better! Still, we can't guarantee we'll find all cuts needed!

Newman and Kcuhta, 2003

Worked on a mining problem, similar to Fricke's but with different objectives. (Paper downloadable from web).

Newman and Kcuhta: Solve aggregated problem, then solve the IP of disaggregated problem.

Me: Made the same mistake as I did, didn't know this was already suggested by Hallefjord and Storoy. But, there are still a lot that could be done.

So, what could be done?

1. The obvious. You could still use branch-and-bound. At each node, use Zipkin's type method to solve the LP. You may then ask: Why don't we simply use the traditional column generation?

2. The not so obvious.

   (a) Under what conditions on matrix coefficients can we guarantee for IPs with equality constraints, that the disaggregation will always work?

   (b) No one has made any conclusions about relations of $\tilde{z}$ and z^* for IPs in general, but you can certainly say something about problem-specific IPs.

   (c) Zipkin's iterative aggregation-disaggregation (IAD) idea is actually good. Though it cannot be directly applied to IPs, developing schemes to IAD for IPs can help you kill time.

But don't be too happy just yet. A direct application of branch-and-bound type scheme may not be so straight forward. Suppose solving the IP form of the Aggregated problem and the IP form of the disaggregation, we *do* get feasible integer solutions. Then create branches on the tree by branching on the values of the y variables seems plausible. It worked for my toy problem! But, the nice feature of decreasing upper bounds deeper down the tree that you can find in normal branch-and-bound methods is not guaranteed to happen in here!

Hence a direction for research!

(d) Hallejford and Storoy's and Litvinchev and Rangel's idea of generating cover cuts for binary IPs using dual solutions from the Aggregated Problem is also good.
Thinking exercise: What about similar things for IPs with general integer variables?

Papers on error bounds

1. P.H. Zipkin, *Bounds on the effect of aggregating variables in linear programs* **Operations Research 28:2** (1980) 403–418.

2. R. Mendelssohn, *Improved bounds for aggregated linear program* **Oper. Res. 28:6** (1980) pp 1450–1453.

3. A. Hallefjord, S. Storoy, *Aggregation and disaggregation in integer programming problems* **Operations Research 38:4** (1990) 619–623.

4. K. Jörnsten, R. Leisten, *Column aggregation and primal decomposition in linear programming: Some observations* **Optimization 24** (1992) 141–156.

5. S.Storoy, *Weights improvement in column aggregation*, **EJOR., Vol 73** (1994).

6. S.Storoy, *Optimal weights and degeneracy in aggregated linear*

programs **Operations Research Letters, Vol. 19** (1996).

7. K.Jörnsten, R.Leisten and S.Storoy, *Convergence aspects of adaptive clustering in variable aggregation*, **Computers & Oper. Res., Vol. 26** (1999).

8. I. Litvinchev, S. Rangel, *Localization of the optimal solution and a posteriori bounds for aggregation*, **Comp. & Oper. Res. 26** (1999) 967–988.

Review paper

1. D.F. Rogers, R.D. Plante, R.T. Wong, J.R. Evans, *Aggregation and disaggregation techniques and methodology in optimization* **Oper. Res. 39:4** (1991) 553–582.

Applications

1. E. Balas, *Solution of large-scale transportation problems through aggregation* **Oper. Res.** (1965) 82–93.

2. R. Mendelssohn, *An iterative aggregation procedure for MDP*, **Oper. Res. 30:1** (1982) 62–73.

3. A. Alverez, O. Chacon, I.S. Litvinchev, and S. Rangel, *Aggregation in the generalized transportation problem*, **Journal of Computer and Systems Sciences International, 40:6** (2001) 923–929.

4. A.M. Newman, M. Kuchta, *Eliminating variables and using aggregation to improve the performance of an integer programming production scheduling model for an underground mine*.

Related papers

1. H. Marhand, L.A. Wolsey, *Aggregation and mixed integer rounding to solve MIPS* **Oper. Res. 49:3** (2001) 363–371.

2. B. Bachelet, *Aggregation approaches for the minimum binary cost tension problem*.

3. O. du Merle, J.-Ph. Vial *Proximal ACCPM, a cutting plane method for column generation and Lagrangian relaxation: application to the p-median problem*.

Aggregation of different kind: Integer Equivalent aggregation

Very interesting, must not miss!

1. F. Glover, D.A. Babayec, **Ann. Oper. Res. 58** (1995) 227–242.

2. D.A. Babayev, F. Glover, J. Ryan **INFORMS Journal on Computing 9:1** (1997) 43–50.

What it's about: Find conditions on w_1, w_2 such that

$$\sum_{j \in N} (w_1 a_{1j} + w_2 a_{2j}) x_j = a_{30},\ x \in X,$$

has same set of solution as:

$$\begin{aligned} \sum_{j \in N} a_{1j} x_j &= a_{10},\ x \in X, \\ \sum_{j \in N} a_{2j} x_j &= a_{20},\ x \in X. \end{aligned}$$

Papers on constraint aggregation or surrogate relaxation

1. I. Elhallaoui, D, Villeneuve, F. Soumis, G. Desaulniers, *Dynamic Aggregation of Set Partitioning Constraints in Column Generation*.

2. Bernard Fortz, *Combinatorial optimization and telecommunications*.

3. L.A.N. Lorena, M.A. Pereira, *A Lagrangean/surrogate heuristic for the maximal covering location problem using Hillsman's edition*.

4. E.L.F. Senne, L.A.N. Lorena, *A Lagrangean/surrogate approach to p-median problems*.

5. L.A.N. Lorena, M.G. Narciso, *Using local surrogate information in Lagrangean relaxation: An application to symmetric traveling salesman problems*.

6. R. Kannan, *Polynomial-time aggregation of IP problems*. **J. of the Assoc. for Comp. Mach. 30:1** (1983) 133–145.