MANAGEMENT SCIENCE
DOI 10.1287/mnsc.1100.1249ec

e-companion

ONLY AVAILABLE IN ELECTRONIC FORM

Electronic Companion—"Maximizing the Efficiency of the U.S. Liver Allocation System Through Region Design" by Nan Kong, Andrew J. Schaefer, Brady Hunsaker, and Mark S. Roberts, *Management Science*, DOI 10.1287/mnsc.1100.1249.

Electronic Companion

Appendix A

An Overview of Branch and Price

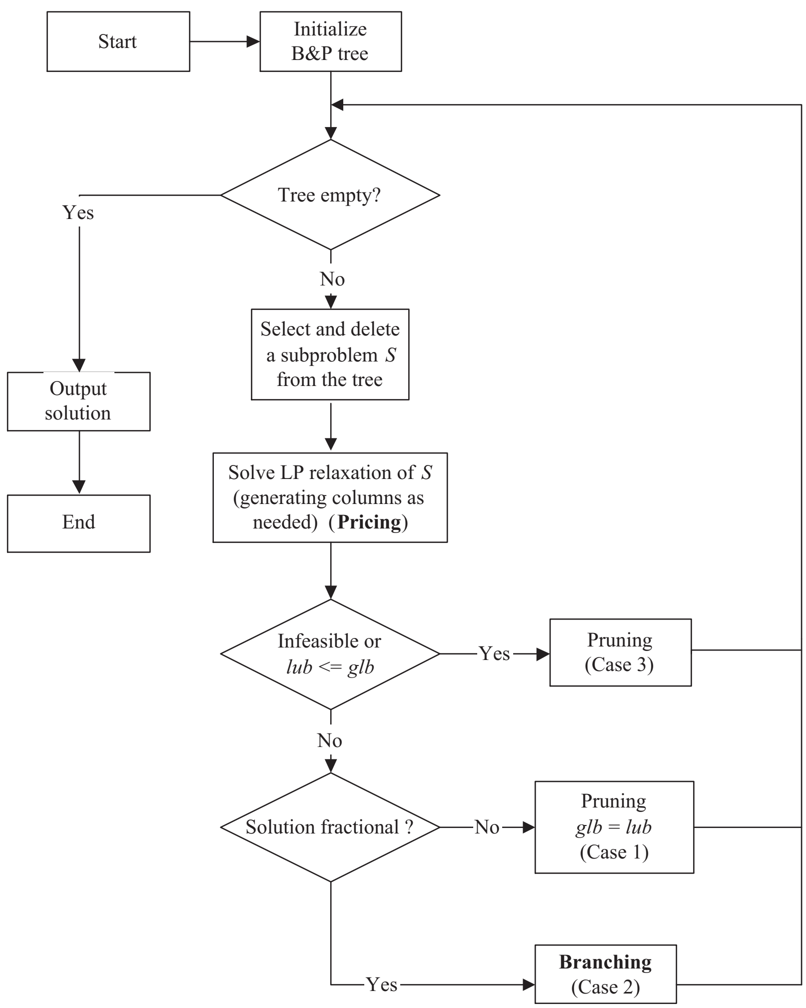

Figure 7 An Overview of Branch and Price

An Algorithmic Description of Branch and Price

Algorithm 1. A Branch-and-Price Algorithm

Input: An optimal region design problem instance; a collection of geographic covers $\mathcal{I}$; and a set of columns R^0 that contains a feasible regional configuration.

Output: An optimal solution x^* to RMP $\left(\bigcup_{i=1}^{|\mathcal{I}|} R_i'\right)$, and its corresponding regional configuration $\mathcal{R}^*$.

Step 1. Generate the initial restricted master problem $\mathcal{S}_{R^0}$. Set $B \leftarrow \{\mathcal{S}_{R^0}\}$ and $\mathcal{L} \leftarrow \emptyset$. Set the global lower bound $glb = 0$ (or $\sum_{r\in\mathcal{R}_0} c_r$ where $\mathcal{R}_0$ is some given regional configuration, i.e., the current configuration).

Step 2. If $B = \emptyset$, set $x^* \leftarrow \hat{x}$ and the corresponding regional configuration $\hat{\mathcal{R}}$ to be $\mathcal{R}^*$. STOP and output x^* and $\mathcal{R}^*$. Otherwise, choose a subproblem $\mathcal{S} \in B$. Set $B \leftarrow B \backslash \{\mathcal{S}\}$. Perform the pricing and bounding operations on $\mathcal{S}$ (see Algorithm 2).

Step 3. If the result of Algorithm 2 is Case 1 (integer solution), then for each $\mathcal{S} \in B$, compare $UB(\mathcal{S})$ with glb. If $UB(\mathcal{S}) \leq glb$, set $B \leftarrow B \backslash \{\mathcal{S}\}$. Go to Step 2.

Step 4. If the result of Algorithm 2 is Case 2 (fractional solution), perform the branching operation on $\mathcal{S}$ (see Algorithm 3). Go to Step 2.

Step 5. If the result of Algorithm 2 is Case 3 (fathomed), go to Step 2.

Algorithm 2. Pricing and Bounding Operations

Input: A subproblem $\mathcal{S}$, described in terms of a subset of potential regions, $R' \subseteq R$; a set of constraints $\mathcal{L}(\mathcal{S})$ modeling the relationships among OPOs; and the global lower bound glb.

Output: Either of the following three cases: (1) an integer optimal solution $\hat{x}$ to the subproblem $\mathcal{S}$ and an updated glb; or (2) an LP relaxation solution $\hat{x}_{LP}(\mathcal{S})$ and the corresponding upper bound $UB(\mathcal{S})$ on the optimal value of $\mathcal{S}$; or (3) otherwise.

Step 1. Set $\hat{R} \leftarrow R'$.

Step 2. Solve $\max\{\sum_{r\in\hat{R}} c_r x_r \mid \sum_{r\in\hat{R}} a_{ir} x_r = 1, \forall i \in I; x_r \geq 0, \forall r \in \hat{R}\}$ and obtain the primal solution $\hat{x}$ and dual solution $\hat{\pi}$.

Step 3. Given a set of constraints on y variables corresponding to $\mathcal{L}(\mathcal{S})$, for all $I_i \in \mathcal{I}$, solve $RPP(\hat{\pi}, I_i)$ to generate $\hat{R}_i'$, one or more new columns that price out favorably. If $RPP(\hat{\pi}, I)$ is infeasible for each $I \in \mathcal{I}$, STOP (Case 3: fathomed by infeasibility).

Step 4. Otherwise, if $\hat{R}' \neq \emptyset$, set $\hat{R} \leftarrow \hat{R} \cup \hat{R}'$ and go to Step 2. Otherwise, $\hat{x}$ is an optimal solution to $\mathcal{S}$. If $\hat{x} \in \mathbb{B}_+^{|\hat{R}|}$, let $\hat{\mathcal{R}}$ be the corresponding regional configuration and go to Step 5. Otherwise, go to Step 6.

Step 5. If $\sum_{r \in \hat{\mathcal{R}}} c_r > glb$, set $glb \leftarrow \sum_{r \in \hat{\mathcal{R}}} c_r$. STOP and output $\hat{x}$ and glb (Case 1). Otherwise, STOP (Case 3: fathomed by integrality).

Step 6. If $\sum_{r \in \hat{R}} c_r \hat{x}_r > glb$, set $\hat{x}_{LP}(\mathcal{S}) \leftarrow \hat{x}$ and $UB(\mathcal{S}) \leftarrow \sum_{r \in \hat{\mathcal{R}}} c_r \hat{x}_r$. STOP and output $\hat{x}_{LP}(\mathcal{S})$ and $UB(\mathcal{S})$ (Case 2). Otherwise, STOP (Case 3: fathomed by bounding).

Algorithm 3. Branching Operation

Input: Subproblem $\mathcal{S}$; corresponding constraint set $\mathcal{L}(\mathcal{S})$; and $\hat{x}_{LP}(\mathcal{S})$.

Output: $\mathcal{S}_1$ and $\mathcal{S}_2$, two children subproblems of $\mathcal{S}$; and $\mathcal{L}(S_1)$ and $\mathcal{L}(S_2)$, the respective corresponding constraint sets of $\mathcal{S}_1$ and $\mathcal{S}_2$.

Step 1. Determine a pair of OPOs (s,t) such that $0 < \sum_{j:a_{sj}=1,a_{tj}=1} \hat{x}_{LP}(\mathcal{S}) < 1$. Then $\mathcal{L}(\mathcal{S}_1) = \mathcal{L}(\mathcal{S}) \cup \{y | y_s = y_t\}$ and $\mathcal{L}(\mathcal{S}_2) = \mathcal{L}(\mathcal{S}) \cup \{y | y_s + y_t \leq 1\}$.

Step 2. Set $B \leftarrow B \cup \{\mathcal{S}_1, \mathcal{S}_2\}$.

Appendix B

Tuning the Branch-and-Price Algorithm

Feasible Columns Generation. Table 2 reports our experiments on the effect of generating multiple feasible columns in addition to the optimal column when solving each pricing problem. Strategy “None” adds no feasible columns to the iterative restricted master problem other than the optimal one. Strategy “All” adds all feasible columns that are identified in the pricing problem solution process to the iterative restricted master problem. These columns include the optimal column for the pricing problem. Strategy i, $i = 1, \ldots, 4$, adds at most the first i feasible columns for each pricing problem.

In the column “CPU Time”, we underline the best CPU time among all considered pricing strategies. In the next two columns, we report the number of restricted master problems solved

Table 2 Effect of Pricing Strategy on CPU Time

	Linear Decay			Cubic Decay		
Pricing Strategy	CPU Time	Num. Iters.	Num. Cols.	CPU Time	Num. Iters.	Num. Cols.
None	0:31:10	25	395	0:33:32	26	365
1	0:56:30	56	621	0:37:30	44	623
2	0:19:08	21	574	0:29:46	27	613
3	0:17:50	18	627	0:22:15	20	646
4	0:21:41	19	607	0:20:10	18	631
All	0:18:01	16	644	0:16:54	15	701

at the root node of the branch-and-price tree and the total number of columns generated at the root node. Among the various pricing strategies, strategies "None" and "1" are always inferior with respect to the solution time. Strategies "3", "4", and "All" are computationally comparable for both viability cases. This observation suggests that allowing 3 or more columns per pricing problem is likely to generate columns that are potentially important for constructing the optimal basis. If it is computationally tractable to solve a pricing problem to optimality, it tends to be preferable to add all columns that can be generated with each pricing problem.

Restricted Master Problem Initialization. Table 3 reports the relative increase of intra-regional transplants over the current regional configuration. We first considered only the $|I|$ single-OPO regions as the set of the initial columns. We call this initialization scheme the "singleton" scheme. We also considered all contiguous 4-OPO regions together with all single-OPO regions as the initial column set. We call this scheme the "All4" scheme. The current regional configuration was also considered as an initial column set. Furthermore, we considered two regional configurations obtained by solving the region design problem through explicit contiguous region enumeration. The two configurations were selected among all contiguous regions with no more than 7 and 8 OPOs, respectively. We call the last 3 initialization schemes the "current", "OPT7", and "OPT8" schemes.

Table 3 indicates that the solution time is relatively insensitive to the initialization scheme. It should be noted that the CPU time in the table for either scheme "OPT7" or "OPT8" does not include the time taken to obtain the respective set of initial columns. Hence, we conclude that the

Table 3 Effect of RMP Initialization

	Linear Decay			Cubic Decay		
Initialization Scheme	Relative Improvement	Num. of Regions	CPU Time	Relative Improvement	Num. of Regions	CPU Time
singleton	8.29%	8	0:18:01	8.46%	9	0:16:54
All4	8.67%	7	0:15:52	8.92%	8	0:17:34
current	8.29%	8	0:21:42	8.46%	9	0:16:10
OPT7	8.29%	8	0:14:16	8.46%	9	0:18:26
OPT8	8.63%	7	0:17:51	8.82%	8	0:18:58

scheme "All4" is the most beneficial among these studied initialization schemes in terms of both solution quality and time.

Initial Geographic Covers Design. Table 4 presents the performance of randomly generated initial geographic covers designs. As a mnemonic, each collection of geographic covers is labeled with m-n, in which m is the number of geographic covers and n is the number of OPOs in each geographic cover. Each of these collections contains 10 to 30 geographic covers and 8 to 12 OPOs in each geographic cover. An implicit specification here is that all geographic covers in any design have the same number of OPOs.

Table 4 Effect of Initial Geographic Covers (I)

	Linear Decay			Cubic Decay		
Geo. Cover Design	Relative Improvement	Num. of Regions	CPU Time	Relative Improvement	Num. of Regions	CPU Time
10-10	3.10%	11	0:07:01	3.26%	11	0:09:02
15-10	6.23%	9	0:31:47	6.40%	10	0:26:48
20-10	8.67%	7	0:15:52	8.92%	8	0:17:34
25-10	9.05%	7	1:34:27	9.15%	8	1:03:16
30-10	9.41%	7	2:40:28	9.51%	8	1:43:17
20-8	3.39%	10	0:02:03	3.55%	11	0:02:08
20-9	6.37%	9	0:04:05	6.60%	9	0:05:20
20-10	8.29%	7	0:15:52	8.46%	8	0:17:34
20-11	9.65%	7	1:08:52	9.84%	8	1:11:06
20-12	10.8%	7	3:12:05	10.9%	8	2:48:04

The table reports the relative increase of intra-regional transplant cardinality compared to the current regional configuration, the number of regions in the terminating regional configuration, and the CPU time (h:mm:ss). Although these experiments depend on the specific geographic covers designs tested, they suggest that the solution time increases as the parameters m and n increase.

Table 5 Effect of Initial Geographic Covers (II)

	Linear Decay			Cubic Decay		
Geo. Cover Design	Relative Improvement	Num. of Regions	CPU Time	Relative Improvement	Num. of Regions	CPU Time
1	8.67%	7	0:15:52	8.92%	8	0:17:34
2	7.96%	8	0:17:47	8.36%	8	0:14:42
3	5.15%	8	0:16:43	5.63%	8	0:19:04
4	7.26%	8	2:22:12	7.45%	9	$\geq$ 5 hrs
5	7.63%	8	$\geq$ 5 hrs	7.50%	9	$\geq$ 5 hrs

On the other hand, the optimal objective value increases as m and n increase. Between m and n, n affects the solution time and quality more. With these tests, we concluded that $m = 20$ and $n = 10$ are a good choice when applying geographic decomposition. The computational results also appear to indicate that the number of regions in a terminating regional configuration does not increase as m and n increases.

We next tested a few randomly generated 20-10 geographic covers designs in addition to the one tested previously. If the algorithm did not terminate after 5 hours, we used the best regional configuration found during the 5-hour solution. We report the computational results in Table 5. Our results show that the choice of geographic covers is critical to the performance of branch and price. A small modification may change the solution significantly. In general, the more overlapping the geographic covers are in a geographic covers design, the higher the solution quality is. The variation across geographic covers led us to integrate branch and price with local search, as described in Section 3.3.

Appendix C

An iterative algorithm framework integrating branch and price and local search

Since our branch-and-price implementation is dependent upon the geographic covers design, it is not easy to guarantee the quality of the terminating regional configuration. To alleviate this solution dependency, we integrate branch and price with local search in an iterative algorithmic framework, presented as follows.

Let ORD($\mathcal{I}$) be the optimal region design problem over $\mathcal{I}$, a collection of geographic covers, and let R'_i be the set of potential regions given $I_i \in \mathcal{I}$ for $i = 1, \ldots, |\mathcal{I}|$. With geographic decomposition, we in fact solve the restricted master problem RMP($\cup_{i=1}^{|\mathcal{I}|} R'_i$). Note that RMP($\cup_{i=1}^{|\mathcal{I}|} R'_i$) is the LP relaxation of ORD($\mathcal{I}$).

Algorithmic Framework

Step 0: Initial Geographic Cover Construction

Construct the initial collection of geographic covers $\mathcal{I}_1$. Set $k = 1$.

Step 1: B&P Algorithm

Apply the branch-and-price algorithm to solve the problem ORD($\mathcal{I}_k$). Let $\hat{x}^k$ be the optimal solution to ORD($\mathcal{I}_k$). Let $\hat{\mathcal{R}}_k$ be the optimal regional configuration, i.e., $r \in \hat{\mathcal{R}}_k$ if $\hat{x}_r^k = 1$.

Step 2: Local Improvement

Set $\mathbf{I}_k$ to be $\emptyset$. Let $r_1^0 = r_1$ and $r_2^0 = r_2$.

For any two distinct regions $r_1, r_2 \in \hat{\mathcal{R}}_k$, perform the following two local search operations.

(a) (Swapping): Let $r_1 = r_1^0$ and $r_2 = r_2^0$. Select OPO $i \in r_1$ and OPO $j \in r_2$. Update $r_1 \leftarrow r_1 \backslash \{i\} \cup \{j\}$ and $r_2 \leftarrow r_2 \backslash \{j\} \cup \{i\}$. Let $\mathcal{R}'$ be the regional configuration after the above modification. If $\sum_{r \in \mathcal{R}'} c_r > \sum_{r \in \mathcal{R}_k} c_r$, then $\mathbf{I}_k = \mathbf{I}_k \cup \{\mathcal{R}'\}$. Repeat the operation until all OPO pairs $(i, j) \in (r_1, r_2)$ have been considered.

(b) (Moving): Let $r_1 = r_1^0$ and $r_2 = r_2^0$. Select OPO $i \in r_1$. Update $r_1 \leftarrow r_1 \backslash \{i\}$ and $r_2 \leftarrow r_2 \cup \{i\}$. Let $\mathcal{R}'$ be the regional configuration after the above modification. If $\sum_{r \in \mathcal{R}'} c_r > \sum_{r \in \mathcal{R}_k} c_r$, then $\mathbf{I}_k = \mathbf{I}_k \cup \{\mathcal{R}'\}$. Repeat the operation until all OPOs $i \in r_1$ have been considered.

Step 3: Stopping Rule Checking

If $\mathbf{I}_k = \emptyset$, then STOP. Otherwise, choose $\mathcal{R}_k^* \in \arg\max_{\mathcal{R}' \in \mathbf{I}_k} \{\sum_{r \in \mathcal{R}'} c_r - \sum_{r \in \hat{\mathcal{R}}_k} c_r\}$ and let r_1^* and r_2^* be the two modified regions, comparing $\mathcal{R}^*$ and $\hat{\mathcal{R}}_k$.

Step 4: Geographic Cover Update

Compare r_1^0, r_2^0 and r_1^*, r_2^* to update $\mathcal{I}_k$ as follows. For each $I_i \in \mathcal{I}_k$, $i = 1, \ldots, |\mathcal{I}_k|$, if $r_j^0 \subseteq I_i$ but r_j^* in not a subset of I_i, then update $I_i \leftarrow I_i \cup r_j$, $j = 1, 2$.

Let $k \leftarrow k + 1$ and go to Step 1.