Submitted to *INFORMS Journal on Computing*
manuscript (Please, provide the mansucript number!)

Online Supplement for Paper: Chance-Constrained Binary Packing Problems

Yongjia Song, James R. Luedtke
University of Wisconsin-Madison, jrluedt1@wisc.edu

Simge Küçükyavuz
The Ohio State University, kucukyavuz.2@osu.edu

1. Calculating $\mathbb{P}(\tilde{A}x \leq \tilde{b})$

Solving formulation (3) of the main text requires being able to evaluate $\mathbb{P}(\tilde{A}x \leq \tilde{b})$ for any $x \in \{0,1\}^n$. This is difficult in general, although it can be seen as a minimal requirement for being able to solve the problem (1) exactly. When $m = 1$, $\mathbb{P}(\tilde{A}x \leq \tilde{b})$ can be calculated efficiently when the coefficients are joint normally distributed or the coefficients are independent and have Poisson distribution. These same examples apply for $m > 1$ provided the coefficients in different rows are independent of each other. See also Song (2013) for an example with $m = 1$ illustrating when $\mathbb{P}(\tilde{A}x \leq \tilde{b})$ can be calculated efficiently provided the coefficients are joint normal with parameters determined by the outcome of a random variable with small finite support.

2. Local cuts

We describe here the details of local cuts for chance-constrained binary packing problems, as described in Section 2.2 of the paper. To obtain a restricted set, we choose disjoint sets $U \subset N$ and $L \subset N$ and fix $x_j = 1, \forall j \in U$ and $x_j = 0, \forall j \in L$. When separating a given solution $\hat{x} \in [0,1]^n$ we choose $U \supseteq N_1 := \{j \in N \mid \hat{x}_j = 1\}$ and $L \supseteq N_0 := \{j \in N \mid \hat{x}_j = 0\}$. We choose U and L such that the size of the set of unrestricted variables $R = N \setminus (L \cup U)$ is at most our limit of 10. If using $U = N_1$ and $L = N_0$ is not sufficient to make R small enough, we include variables j in U that have the largest $\hat{x}_j$ values and variables $j \in L$ that have the smallest $\hat{x}_j$ values. Let $\tilde{A}^R$ be the submatrix of $\tilde{A}$ with columns corresponding to R, and $\tilde{b}^R := \tilde{b} - \sum_{j \in U} \tilde{A}_{\cdot j}$. Then the restricted set is:

$$X(L,U) = \left\{x \in \{0,1\}^{|R|} \mid \mathbb{P}(\tilde{A}^R x \leq \tilde{b}^R) \geq 1 - \epsilon\right\}$$

Let $\hat{x}_R$ be the subvector of $\hat{x}$ corresponding to the variables in the set R. We solve a polar linear program to try to obtain an inequality valid for $X(L,U)$ that cuts off $\hat{x}_R$. To obtain this linear program, we enumerate the set of all maximal packs in the set $X(L,U)$, which we denote by $\bar{x}^h, h \in H$. The size of this set grows exponentially in $|R|$, which is why we restrict R to be small. Since $X(L,U)$ is a down-monotone set, all nontrivial valid inequalities will have nonnegative coefficients. We therefore solve the dual of the following polar linear program:

$$\max \sum_{j \in R} \hat{x}_j \alpha_j - \beta \tag{OS-1a}$$

$$\text{s.t.} \sum_{j \in R} \bar{x}_j^h \alpha_j - \beta \leq 0, \ \forall h \in H \tag{OS-1b}$$

$$\sum_{j \in R} \alpha_j = 1 \tag{OS-1c}$$

$$\beta \geq 0, \alpha_j \geq 0, \forall j \in R \tag{OS-1d}$$

where (OS-1c) is a normalization constraint on the coefficients α. A cut separating $\hat{x}_R$ from the convex hull of $X(L,U)$ exists if and only if the optimal value of (OS-1) is positive.

In our implementation, we explicitly enumerate the maximal packs in $X(L,U)$. Although we did not explore this option, another approach (e.g., (Applegate et al. 2006)), is to solve the dual of (OS-1) via column generation, in which case the column generation subproblem would have the form of a (smaller) chance-constrained packing problem.

If an inequality $\hat{\alpha}x_R \leq \hat{\beta}$ that cuts off $\hat{x}_R$ is identified, we then obtain an integer vector α' that closely approximates $\hat{\alpha}$, using the continued fraction approach presented in (Applegate et al. 2006). We then recompute a valid right-hand side $\beta' = \max_{h\in H} \alpha'\bar{x}^h$. Then, beginning with the inequality $\alpha' x_R \leq \beta'$, we downlift the variables in the set U to obtain a valid inequality for the unrestricted feasible region (which may not necessarily cut off the solution $\hat{x}$). Finally, we up-lift the variables in the set L.

3. Proof of Theorem 1

Recall that Z is the set of $x \in [0,1]^n, z \in [0,1]^{|S|}$ that satisfy:

$$\sum_{k\in S} p_k z_k \geq 1-\epsilon \tag{OS-2}$$

$$A^k x \leq b^k + M^k(1-z_k), \forall k \in S. \tag{OS-3}$$

Theorem 1. $\text{proj}_x(Z)$ *is described by* $x \in [0,1]^n$ *and the inequalities:*

$$A^k x \leq b^k + M^k, \quad \forall k \in S \tag{OS-4}$$

$$\sum_{k\in\bar{S}} \frac{p_k}{M^k_{\psi(k)}}(A^k_{\psi(k)\cdot}x - b^k_{\psi(k)}) \leq \epsilon, \qquad \forall \bar{S} \subseteq S, \psi \in \Psi(\bar{S}). \tag{OS-5}$$

Proof. "$\subseteq$": Suppose $\hat{x} \in \text{proj}_x(Z)$. Then $\hat{x} \in [0,1]^n$ and $\exists \hat{z} \in [0,1]^{|S|}$, such that (OS-2) and (OS-3) hold. Because $\hat{z}_k \geq 0$, (OS-4) follows from (OS-3).

Given any $\bar{S} \subseteq S$, and any mapping $\psi \in \Psi(\bar{S})$, by aggregating the inequalities $A^k_{\psi(k)\cdot} x \le b^k_{\psi(k)} + M^k_{\psi(k)}(1 - z_k)$ for $k \in \bar{S}$ with weights $p_k / M^k_{\psi(k)}$, we have:

$$\sum_{k \in \bar{S}} \frac{p_k}{M^k_{\psi(k)}} (A^k_{\psi(k)\cdot} \hat{x} - b^k_{\psi(k)}) \le \sum_{k \in \bar{S}} p_k (1 - \hat{z}_k) \le \sum_{k \in S} p_k (1 - \hat{z}_k)$$
$$= 1 - \sum_{k \in S} p_k \hat{z}_k \le \epsilon$$

and so (OS-5) holds as well.

"$\supseteq$": Let $\hat{x} \in [0,1]^n$ satisfy (OS-4) and (OS-5). In particular, let $\bar{S} = \big\{ k \in S \mid A^k_{i\cdot} \hat{x} > b^k_i \text{ for some } i \in \{1, \ldots, m\} \text{ with } M^k_i > 0 \big\}$ and choose $\psi \in \Psi(\bar{S})$ by $\psi(k) \in \arg\max_{i=1,2,\ldots,m} \{ (A^k_{i\cdot} \hat{x} - b^k_i)/M^k_i \mid M^k_i > 0 \}$. Then, with $\alpha^k = A^k_{\psi(k)\cdot}, \beta^k = b^k_{\psi(k)}$, and $\mu^k = M^k_{\psi(k)}$, by (OS-5) we have:

$$\sum_{k \in \bar{S}} \frac{p_k}{\mu^k} (\alpha^k \hat{x} - \beta^k) \le \epsilon. \tag{OS-7}$$

Now let $\hat{z}$ be defined as :

$$\hat{z}_k = \begin{cases} 1 & k \in S \setminus \bar{S} \\ 1 - (\alpha^k \hat{x} - \beta^k)/\mu^k & k \in \bar{S}. \end{cases}$$

We claim that $(\hat{x}, \hat{z}) \in Z$. For $k \in \bar{S}$, we know that $0 < \alpha^k \hat{x} - \beta^k \le \mu^k$ by (OS-4), thus $(\alpha^k \hat{x} - \beta^k)/\mu^k \in (0,1], \forall k \in \bar{S}$, so $\hat{z} \in [0,1]^{|S|}$.

We next show that (OS-3) holds. First, if $k \in S \setminus \bar{S}$, $A^k \hat{x} \le b^k = b^k + M^k(1 - \hat{z}_k)$ by construction of $\bar{S}$. If $k \in \bar{S}$, then $\hat{z}_k = 1 - (\alpha^k \hat{x} - \beta^k)/\mu^k$ thus for each component $i = 1, 2, \ldots, m$ such that $M^k_i > 0$:

$$A^k_{i\cdot} \hat{x} - b^k_i - M^k_i (1 - \hat{z}_k) = A^k_{i\cdot} \hat{x} - b^k_i - M^k_i (\alpha^k \hat{x} - \beta^k)/\mu^k$$
$$= [(A^k_{i\cdot} \hat{x} - b^k_i)/M^k_i - (\alpha^k \hat{x} - \beta^k)/\mu^k] M^k_i \le 0$$

because:

$$(\alpha^k \hat{x} - \beta^k)/\mu^k \ge (A^k_{i\cdot} \hat{x} - b^k_i)/M^k_i, \ \forall i = 1, 2, \ldots, m \text{ s.t. } M^k_i > 0$$

based on the definition of α^k, β^k and μ^k. Also, for each $i = 1, 2, \ldots, m$ such that $M_i^k = 0$ (OS-3) holds by directly by (OS-4).

Finally,

$$\begin{aligned}\sum_{k\in S} p_k \hat{z}_k &= \sum_{k\in \bar{S}} p_k \hat{z}_k + \sum_{k\in S\setminus \bar{S}} p_k \hat{z}_k \\ &= \sum_{k\in \bar{S}} p_k \hat{z}_k + \sum_{k\in S\setminus \bar{S}} p_k = 1 + \sum_{k\in \bar{S}} p_k(\hat{z}_k - 1) \\ &= 1 - \sum_{k\in \bar{S}} \frac{p_k}{\mu_k}(\alpha^k \hat{x} - \beta^k) \geq 1 - \epsilon,\end{aligned}$$

by (OS-7), so (OS-2) is satisfied. □

4. Extended probabilistic cover inequalities

A simple idea for approximately lifting probabilistic cover inequalities is to adapt the idea of the extended cover inequalities used for the deterministic knapsack problem (Nemhauser and Wolsey 1988) to the probabilistic setting.

Definition 1. Given a probabilistic cover C, let F^C be the set of scenarios in which C is a cover. An extended probabilistic cover $E(C)$ is defined as $E(C) := C \cup \{j \in N \mid \exists r \in R(k) \text{ with } A_{rj}^k \geq A_{rj'}^k, \forall j' \in C, k \in F^C\}$, where $R(k) := \{i \in \{1, 2, \ldots, m\} \mid \sum_{j\in C} A_{ij}^k > b_i^k\}$.

Proposition 1. *The following extended probabilistic cover inequality is valid for X:*

$$\sum_{j\in E(C)} x_j \leq |C| - 1. \tag{OS-9}$$

Proof. *By definition, $\forall k \in F^C$, there exists $r \in R(k)$ such that $\sum_{j\in C'} A_{rj}^k \geq \sum_{j\in C} A_{rj}^k > b_r^k$ for each $C' \subseteq E(C)$ with $|C'| = |C|$. Therefore,* (OS-9) *is valid.* □

Clearly, inequality (OS-9) dominates the probabilistic cover inequality with the same cover C when $E(C) \supsetneq C$. However, in our computational experience, we rarely found cases where $E(C) \neq C$. The reason is that, even in the case of an individual knapsack, in order for an element $j \notin C$ to be in $E(C)$, the coefficient A_j^k on element j must be at least as large as that on all elements in C *for all* scenarios in the set F^C.

5. An example for illustrating the heuristic sequential lifting problem

Consider an instance of the finite scenario approximation of individual chance-constrained knapsack problem with four items, a deterministic capacity $b = 52$, and three equally likely item weight scenarios: $a^1 = (20, 30, 40, 1), a^2 = (30, 30, 20, 1), a^3 = (10, 40, 30, 1)$. Let $\epsilon = 0$ so that we do not allow the capacity to be violated in any scenario. Then $C = \{1, 2\}$ is a minimal probabilistic cover, and thus $x_1 + x_2 \leq 1$ is a probabilistic cover inequality. Suppose we use scenario-based heuristic sequential lifting, first on x_3. Then we obtain $\zeta_3(1) = 0, \zeta_3(2) = 1$ and $\zeta_3(3) = 1$. Since $\epsilon = 0$, $\beta_3' = 1 - \zeta_3(\sigma_1) = 1$, and we obtain: $x_1 + x_2 + x_3 \leq 1$. Next we lift variable x_4 and obtain $\zeta_4(1) = 2, \zeta_4(2) = 2$ and $\zeta_4(3) = 2$. Thus $\beta_4' = 1 - \zeta_4(\sigma_1) = -1 < 0$. In this case we use $\beta_4 = 0$.

We also see from this example that $\zeta_4(1) = 2 > |C| - 1 = 1$, and thus to calculate $\zeta_4(1)$ using the expression $\zeta_4(1) = \max\{y : \eta_4^1(y) \geq b^1 - a_4^1\}$ we would need to have calculated $\eta_4^1(y)$ for $y = 2 > |C| - 1$. Also, for deterministic binary knapsack problems $\eta_{\pi_{j+1}}(y) = \eta_{\pi_j}(y)$ always holds when $y < \beta_{\pi_j}$, since in that case $\eta_{\pi_j}(y) \leq a_{\pi_j}$. However, in our example, we see that when $y = 1$, $\eta_3^2(1) = \min\{30x_1 + 30x_2 \mid x_1 + x_2 \geq 1, x_1, x_2 \in \{0, 1\}\} = 30 > a_3^2 = 20$, which shows $\eta_{\pi_j}(y) \leq a_{\pi_j}$ does not always hold in the case of scenario-based heuristic sequential lifting.

6. Further implementation details for heuristic sequential lifting

Cover initialization: At each relaxation solution $\hat{x}$, we initialize the cover C in (3b) by including items that correspond to positive relaxation values in the following greedy way. First, we include item j that has current relaxation value $\hat{x}_j = 1$ in the initial probabilistic cover C. Following the notation in (Gu et al.

1998), we call this set $C_2 := \{j \in N \mid \hat{x}_j = 1\}$. If C_2 is not a probabilistic cover yet, we continue to include items in cover C sequentially with a nonincreasing order of $\hat{x}$ until we have a probabilistic cover. The resulting cover may not be minimal. Next we describe a procedure to make it minimal.

Cover reduction: The cover reduction procedure is done sequentially to make the probabilistic cover minimal. The sequence is chosen by an increasing order of relaxation value $\{\hat{x}_j\}_{j=1}^n$, since we would like to first remove items that correspond to smaller relaxation values $\hat{x}$, so that probabilistic cover inequality (3b) is more likely to be violated by $\hat{x}$. Following the sequence, we check if item $j \in C$ is removed, $C \setminus \{j\}$ remains a probabilistic cover. In particular, we do not remove items that correspond to components j with relaxation value $\hat{x}_j = 1$.

Heuristic sequential lifting: We apply the "default" lifting sequence of (Gu et al. 1998) for lifting cover inequalities for deterministic 0-1 knapsack problem. Let the cover $C = C_1 \cup C_2$, where $C_2 = \{j \in C \mid \hat{x}_j = 1\}$, and $C_1 = \{j \in C | \hat{x}_j < 1\}$. Let F be the set of items that are not in the cover, and have positive relaxation values: $F = \{j \notin C \mid 0 < \hat{x}_j < 1\}$. Let W be the set of items that are not in the cover, and have relaxation value 0: $W = \{j \notin C \mid \hat{x}_j = 0\}$. Based on our initialization of the probabilistic cover C, the set of items N is partitioned into $N = C_1 \cup C_2 \cup F \cup W$. If $|C_1| = 0$, we lift from the probabilistic cover inequality $\sum_{j \in C} x_j \le |C| - 1$, and perform a heuristic sequential uplifting on set F and W. Otherwise, we lift from the base inequality

$$\sum_{j \in C_1} x_j \le |C_1| - 1, \tag{OS-10}$$

which is valid when we fix $x_j = 1, \forall j \in C_2$, and is facet-defining for the convex hull of the restricted set with $x_j = 1, \forall j \in C_2$, and $x_j = 0, \forall j \in N \setminus C$.

We first perform a heuristic to sequentially uplift the variables in set F. Within set F, we greedily choose the sequence of lifting in a decreasing order of the corresponding relaxation values $\hat{x}_j$, since the earlier a variable is lifted, the larger its lifting coefficient will be (Gu et al. 1998), and a variable that has higher relaxation value may be better coupled with a higher lifting coefficient in order to obtain a larger violation. Notice that we lift the variables in F while the variables in C_2 are fixed to one, thus we are likely to get better lifting coefficients for variables in F than the case if we simply lift probabilistic cover inequalities. After this procedure, we have the following lifted inequality

$$\sum_{j \in C_1} x_j + \sum_{j \in F} \alpha_j x_j \leq |C_1| - 1, \tag{OS-11}$$

which is valid if $x_j = 1, \forall j \in C_2$.

If the resulting lifting inequality (OS-11) is not violated by the current solution $\hat{x}$, we can abandon the lifting procedure, since downlifting on C_2 only makes the lifted inequality valid, and it will not increase the violation. Otherwise we downlift variables in C_2. Since variables in C_2 all have the same relaxation values (one), we downlift these variables in a lexicographic sequence. Heuristic sequential downlifting is almost identical to heuristic sequential uplifting, and it gives an upper bound on the lifting coefficients. We no longer fix a variable x_j to one once it has been downlifted. After this procedure, we have the following valid lifted inequality

$$\sum_{j \in C_1} x_j + \sum_{j \in F} \alpha_j x_j + \sum_{j \in C_2} \beta_j x_j \leq |C_1| - 1 + \sum_{j \in C_2} \beta_j. \tag{OS-12}$$

At this point, the inequality (OS-12) is a valid inequality. If it is violated, we can add the valid inequality without lifting variables in set W. Notice that lifting variables in W does not help to make the lifted inequality more violated by the current solution, but it may yield a stronger valid inequality that is more useful to cut solutions in which $x_j > 0$ for some $j \in W$.

7. Instance generation details

The individual knapsack instances are obtained by extracting one row at a time from the multi-dimensional deterministic instances 1-7 and weish26. Instance 1-7-1 is obtained from the first row of instance 1-7, instance 1-7-5 is obtained from the fifth row of instance 1-7. Similarly for instances weish26-1 and weish26-5. The set packing instances are transformed from the set partitioning instances sppnw-15 and sppnw-41, by turning the equality constraints $Ax = 1$ into $Ax \leq 1$. We also truncated instance sppnw-15 to have only the first 200 items, whereas there are 467 items in the original deterministic set partition instance. For all the instances, we assume that items may fail to appear according to a Bernoulli distribution with failure probability $\mu_j, \forall j = 1, 2, ..., n$. These failure probabilities μ_j are generated according to an exponential distribution with mean 0.1, and then truncated to be between zero and one. For both individual and multi-dimensional knapsack instances, if an item fails to appear in one scenario, the corresponding item weight in that scenario is zero, otherwise it follows a normal distribution with mean value equal to the item weight in the deterministic instance, and the standard deviation equal to 0.1 times the item weight. We set the item weight value to be zero if the realization of the normal random variable is negative. For set packing instances, we again assume each column of the random 0-1 matrix $\tilde{A}$ randomly fails to appear according to a Bernoulli distribution. We apply exactly the same settings above for generating the failure probabilities.

8. Big-M coefficient strengthening with multiple rounds of iterative LP.

For the instances given in Table 1 of the main paper, we also experimented with doing two rounds of coefficient strengthening using the iterative LP approach of (Qiu et al. 2013). Table 1 shows the results using one and two iterations. We see from Table 1 that solving an additional iteration of the LP-based coefficient strengthening procedure has very little incremental benefit in terms of the big-M coefficient reduction for these instances. The average root gap is only slightly improved in the second iteration, and this leads to only a slight reduction in

Table 1 **Average time spent on strengthening the big-M coefficients, average computational time for solving the MIP, and average root optimality gap for the extended formulation strengthened by one iteration, and two iterations of iterative LP.**

Instances		IterLP-1			IterLP-2		
Instance	$\|S\|$	AvS	AvT	AvR	AvS	AvT	AvR
1-7-1	100	1.3	0.2	2.4%	3.3	0.2	2.2%
	500	79.4	15.3	2.4%	204.2	9.5	2.1%
	1000	573.5	135.9	2.4%	1628.7	71.3	2.1%
1-7-5	100	1.9	0.3	2.1%	2.6	0.2	2.0%
	500	74.9	71.7	2.4%	203.5	41.5	2.1%
	1000	526.6	630.3	2.4%	1668.4	483.4	2.1%
weish26-1	100	2.4	0.4	0.7%	4.4	0.4	0.6%
	500	127.2	101.7	0.9%	349.0	98.2	0.8%
	1000	898.6	2215.9	0.9%	2664.0	2193.8	0.8%
weish26-5	100	2.5	0.4	0.7%	4.1	0.3	0.7%
	500	130.7	84.0	1.0%	324.6	54.4	1.0%
	1000	728.6	1955.6	1.1%	2501.4	1888.9	1.0%

the solution time of the extended formulation. On the other hand, as expected, doing two iterations significantly increases the amount of time spent calculating the improved coefficients.

9. Cut generation computational results

In Table 2 we summarize the average cut generation time, and the average percentage of each type of cut among all cuts generated: lifted cover and lifted pack inequalities, projection cuts (OS-4) and (OS-5), and local cuts, in our best probabilistic cover implementation (referred to as "best prob-cover" in Table 3 of the main text). "AvC" represents the average total number of cuts generated, "ProjT" represents the time spent on generating projection cuts, and "% Proj" represents the percentage of all cuts generated that are projection cuts. Similar notation is used for local cuts, and lifted cover and pack inequalities (which are reported together under "LiftT and "% Lift"). We find that the projection cuts incur little computational burden in terms of number of cuts added and time spent generating them. Lifted cover and pack inequalities require more computational time, but the time does not grow too much as the number of items n grows. Local cuts require the most time, especially on the instances with $n = 90$ items. However, we also see that on these instances the majority of cuts are local cuts, suggesting that it is worth spending this time on these instances.

Table 2 **Detailed summary of the average cut generation time and average percentage of each type of cuts among all cuts generated.**

Instance	$\lvert S\rvert$	AvT	AvN	AvC	ProjT	% Proj	LocalT	% Local	LiftT	% Lift
1-7-1	100	19.2	1544	1592	0.1	2.5%	8.8	16.3%	2.2	81.2%
$n = 50$	1000	81.2	1423	1630	1.0	0.6%	46.3	22.3%	25.4	77.1%
	3000	350.6	1889	2339	4.8	0.5%	211.8	25.8%	113.7	73.8%
1-7-5	100	8.8	869	1368	0.1	1.8%	4.0	16.6%	1.5	81.6%
$n = 50$	1000	47.3	602	1688	0.8	1.7%	22.5	16.5%	18.9	81.8%
	3000	152.0	696	1708	2.4	3.3%	81.5	20.4%	60.4	76.3%
weish26-1	100	20.8	404	1529	0.2	34.6%	12.8	45.7%	1.6	19.8%
$n = 90$	1000	148.7	431	1442	1.8	8.3%	117.2	68.2%	20.9	23.5%
	3000	651.9	616	1983	8.9	5.0%	517.3	70.3%	100.0	24.7%
weish26-5	100	95.7	1410	3628	0.8	18.8%	55.8	57.9%	5.5	23.2%
$n = 90$	1000	696.0	2236	5154	7.9	0.8%	493.2	68.7%	101.2	30.5%
	3000	1612.3	1883	4229	21.2	0.6%	1242.9	69.5%	252.9	29.9%

References

Applegate, D.L., R.E. Bixby, V. Chvátal, W.J. Cook. 2006. *The Traveling Salesman Problem*. Princeton University Press, Princeton, NJ.

Gu, Z., G.L. Nemhauser, M.W.P. Savelsbergh. 1998. Lifted cover inequalities for 0-1 integer programs: Computation. *INFORMS J. Comput.* **10** 427–437.

Nemhauser, George L., Laurence A. Wolsey. 1988. *Integer and Combinatorial Optimization*. Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley.

Qiu, F., S. Ahmed, S. Dey, L. Wolsey. 2013. Covering linear programs with violations. *INFORMS J. Comput.* Accepted for publication.

Song, Y. 2013. Structure-exploiting algorithms for chance-constrained and integer stochastic programs. Ph.D. thesis, University of Wisconsin-Madison.