문제 특성과 알고리듬 수행 능력 간 관계에 관한 분석 : 0-1 Knapsack 문제에 관한 사례 연구

†양재환* · 김현수**

An Analysis of the Relationship between Problem Characteristics and Algorithm Performance : A Case Study on 0-1 Knapsack Problems

Jaehwan Yang* · Hyunsoo Kim**

■ Abstract ■

We perform a computational study on 0-1 knapsack problems generated under explicit correlation induction. A total of 2000 100-variable test problems are solved. We use two solution methods : (1) a well known heuristic and (2) a representative branch and bound type algorithm. Two different performance measures are considered : (1) the number of nodes needed to find an optimal solution and (2) the relative error of the heuristic solution. We also examine the effect of different joint probability mass functions (pmfs) for the coefficient values on the performance of the solution procedure.

Keyword : Test Problem Generation, 0-1 Knapsack Problem, Computational Testing of Algorithm/Heuristics, Explicitly Correlated Induction

논문접수일 : 2005년 4월 1일 논문게재확정일 : 2006년 1월 16일

* 서울시립대학교 경영학부
** 경기대학교 첨단산업공학부
† 교신저자

1. Introduction

When generating test problems to evaluate solution methods for optimization problems, it is common to assume that the coefficient types are mutually independent and the marginal distribution of values for each coefficient type is discrete uniform [27]. Moore [16] and Reilly [25] address shortcomings of these two assumptions. For example, Reilly [25] points out that the coefficient in actual instances of optimization problems may not be probabilistically independent or uniformly distributed. In addition, Moore [16] suggests that the test problems should be hard enough to challenge many solution procedures so that an accurate assessment of the procedures' potential can be made. Test problems whose coefficients are generated under independently are often moderately easy problems. Hooker [5, 6] argues that an empirical science of algorithms is required for testing algorithms. He also points out that to capture insight on algorithm performance, the experimental rigor of computational testing must be improved.

To overcome shortcomings in common practices in the empirical testing of solution procedures for discrete optimization problems, Peterson and Reilly [23] and Reilly [26, 28] suggest a new way to generate test problems with two types of coefficients based on the parametric envelope and parametric mixtures for a bivariate discrete random variable. No matter what kind of discrete probability distributions we have for the objective function and constraint coefficients, their method can generate a test problem with a specified target correlation between objective function and constraint coefficients for problems such as the 0-1 knapsack problem, the weighted set covering problem, and the generalized assignment problem.

Earlier computation studies have shown that the difficulty of a 0-1 knapsack problem, as measured by the number of iterations with a branch and bound procedure, increases as the population correlation between the objective function and constraint coefficient increases[1, 4, 8-12, 16, 18, 25]. By controlling the population correlation between coefficient types, it may be possible to generate either a hard or an easy problem on demand.

Hill and Reilly [7] study the effects of coefficient correlation structure in two-dimensional knapsack problems on solution procedures. In their study, population correlation structure among two dimensional knapsack problems coefficients, the level of constraint slackness, and type of correlations are varied. The CPLEX, version 2.1 is used as a representative branch and bound solution procedure. Cario et al. [3] perform a similar study on the general assignment problems, and different solution procedures are tested including CPLEX.

In this research, we perform a computational study on the 0-1 knapsack problem. Even though the structure of the problem is the simplest among considered, many studies examine the problem due to variety of its applications. Especially, most of them develop efficient algorithms including heuristics [1, 4, 8-14, 20, 24, 29, 30]. While all the studies perform deductive analysis to show their algorithms' superiority to others, test problems are generated without careful consideration for the correlation structure of the 0-1 knapsack problem.

For example, Martello and Toth [8-12] compare the performance of solution methods for the

0-1 knapsack problem on test problems in which the objective function and constraint are uncorrelated (independent), weakly correlated, and strongly correlated. The weakly correlated and strongly correlated problems are generated with the so-called implicit correlation induction. For example, to generate test problems, Martello and Toth [11] use the following forms :

- Uncorrelated problems :
  $A \sim U\{1,2,\cdots,100\}$, $C \sim U\{1,2,\cdots,100\}$,
- Weakly correlated problems :
  $A \sim U\{1,2,\cdots,100\}$, $C \sim U\{-10,-9,\cdots,+10\}+A$,
- Strongly correlated problems :
  $A \sim U\{1,2,\cdots,100\}$, $C = A+10$.

According to Moore and Reilly [22], the population correlation for the weakly correlated problems is more than 0.97, and the coefficients are perfectly correlated in the strongly correlated problems. Balas and Zemel [1] use the same idea as Martello and Toth [11] to generate 0-1 knapsack problems. Martello et al. [13, 14] and Pisinger [24] use the similar idea to generate their 0-1 knapsack problems. Earlier versions of research use even a simpler method to generate the test problems [20, 30].

Reilly [25] introduces a new method to randomly generate 0-1 knapsack problems with a specific target correlation, and Peterson and Reilly [23] and Reilly [26, 28] refine the method by Reilly [25] and suggest a way to generate test problems with two types of coefficients based on the parametric envelope and parametric mixtures for a bivariate discrete random variable. Reilly [25] also solves 200 25-variable 0-1 knapsack problems with specific target correlations between objective function and constraint coefficients such as 0, -1, and +1 by using a branch and bound algorithm. There exist 23 test problems which can not be solved by the algorithm due to exceeding the maximum number of iterations set by the algorithm. He assumes $A \sim U\{1,2,\cdots,50\}$, $C \sim U\{1,2,\cdots,100\}$ and $b = \lfloor \sum_{j=1}^{25} a_j/2 \rfloor$.

In this study, we use the method suggested by Reilly [28] to generate 2000 100-variable 0-1 knapsack problems with different target correlations such as -1, -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75, and 1 between objective function and constraint coefficients. Furthermore, we carefully generate test problems with other parameters such as different joint pmfs and ranges of coefficient values. Then, we examine the results whether correlation structure and other parameters have impact on algorithms we test. We expect that effect of coefficient correlation structure is bigger than that in previous studies due to the simple structure of the 0-1 knapsack problem. The 0-1 knapsack problem has one only constraint, and the correlation of an objective function coefficient to a constraint coefficient seems to be strongly related to determine whether the associated decision variable has a value of 1. Also, by having only one constraint, a concern about existence of correlations between coefficients in different constraints can be removed.

The research is concerned with three possible indicators of problem hardness : (1) the population correlation between the objective function and constraint coefficient, (2) sample correlation between the objective function and constraint coefficient, and (3) the smallest joint probability for any possible coefficient combination. The two solution procedures are used to solve all of the test problems : (1) a well known heuristic and (2) a representative branch and bound algorithm, commercially available CPLEX, version 8.0.

The performance measures we use to evaluate the solution method are : (1) the number of branch and bound nodes from CPLEX needed to find an optimal solution and (2) the relative error of the heuristic solution.

In the next section, we review some background. Then, we specify assumptions and describe the types of test problems that we generate with different combinations of parameters. A basic algorithm to generate test problems and the algorithm we used to solve the test problems are described in Section 5. Computational results and analysis are presented in Section 6. The paper concludes with a summary of the results and limitations of this study.

2. Background

All basic ideas and techniques used in this computational experiment are based on Reilly [25-28], Moore and Reilly [18], and Peterson and Reilly [23]. We review two important concepts from those papers briefly. They are the parametric envelope for a bivariate discrete random variable (A, C), sampling with explicitly induced correlation using parametric mixtures.

2.1 Parametric Envelope for (A, C)

We summarize the idea of a parametric envelope for a bivariate discrete random variable from Peterson and Reilly [23].

Let $F_1(a)$ and $F_2(c)$ be the distribution of a discrete random variable A with a finite support $S_A = \{a_1, a_2, \cdots, a_{n_1}\}$ for $a_1 < a_2 < \cdots < a_{n_1}$ and the distribution of a discrete random variable C with finite support $S_C = \{c_1, c_2, \cdots, c_{n_2}\}$ for $c_1 < c_2 < \cdots < c_{n_2}$, respectively. Also, let θ be the largest possible value of the smallest joint probability over the bivariate support $S_A \times S_C$ and let $\rho = \mathrm{Corr}(A, C)$. Then, Peterson and Reilly [23] show that a curve that plots θ, which is a function of ρ, can be constructed by following the solution of a parametric linear program.

The points on and under the parametric curve, including the horizontal axis represent the set of points that corresponds to all feasible combinations of ρ and θ. Peterson and Reilly [23] call this set of points the parametric envelope for (A, C). At least one pmf for (A, C) is associated with each point in the parametric envelope. Our computational experiment is based on this parametric envelope concept. Specifically, we choose 25 points in the parametric envelope and use them as the target combinations of ρ and θ for our computational experiment.

2.2 Sampling with Explicitly Induced Correlation Based on the Parametric Mixtures

There are at least three methods for generating synthetic optimization problems : under the assumption of mutual independence, with implicitly induced correlation, and with explicitly induced correlation. The assumption of mutual independence has at least two critical shortcomings : the coefficients in actual instances of optimization problems may not be probabilistically independent and the test problems generated under independence may not be hard enough to provide a sufficient challenge for solution methods.

An alternative way to generate test problems is to generate problems in which correlation is implicitly induced between the coefficient types. Moore and Reilly [18] use the term implicit induction of correlation to describe any test problem generation method in which a target population correlation is implied by specifying a functional relationship between coefficient types. They also refer to any generation method in which target population is prespecified as explicit correlation induction.

Reilly [27] points out that the synthetic problems whose coefficients are generated independently are too similar to one another to provide a sufficiently diverse collection of test cases, particularly when the synthetic problems are large. He shows that this shortcoming can be overcome by explicitly inducing correlation between the coefficient types for a variety of population correlation targets.

We can characterize a pmf for (A,C) when a target population correlation, ρ_0, for (A,C) is specified by mixing values of (A,C) generated under independence and values of (A,C) generated with extreme correlation, $\rho_{\max}$ or $\rho_{\min}$, the maximum and minimum possible values for ρ, respectively. The following is the form of these conventional mixtures :

$$g(a,c|\rho_0)=\begin{cases}(1-\rho_0/\rho_{\max})f_1(a)f_2(c)+(\rho_0/\rho_{\max})g_{\max}(a,c) \\ \qquad\qquad\qquad if\ \ \rho_0\geq 0; \\ (1-\rho_0/\rho_{\max})f_1(a)f_2(c)+(\rho_0/\rho_{\max})g_{\min}(a,c) \\ \qquad\qquad\qquad if\ \ \rho_0<0; \end{cases}$$

where $f_1(a)$ and $f_2(c)$ are the marginal pmfs for A and C, $g_{\max}=(a,c)$ is the maximum correlation pmf for (A,C), and $g_{\min}=(a,c)$ is the minimum correlation pmf for (A,C). See Moore [15, 16] for its application.

Peterson and Reilly [23] suggest an alternative to conventional mixtures (1). Let $i^*=\mathrm{argmin}_i\{f_1(a_i)\}, j^*=\mathrm{argmin}_j\{f_2(c_j)\}$, and $\theta^*=f_1(a_{i*})f_2(c_{j*})$.

Consider.

$$\begin{aligned}g(a,c|\rho_0,\theta_0)&=\lambda_0 f_1(a)f_2(c)+\lambda_{\min}g_{\min}(a,c)\\&=\lambda_{\max}g_{\max}(a,c),\end{aligned}\qquad(2)$$

where

$\lambda_0=\theta_0/\theta^*$, $\lambda_{\min}=((1-\theta_0/\theta^*)\rho_{\max}-\rho_0)/(\rho_{\max}-\rho_{\min})$, and $\lambda_{\max}=(\rho_0-(1-\theta_0/\theta^*)\rho_{\min})/(\rho_{\max}-\rho_{\min})$ where θ_0 is a target value of θ. If (ρ_0,θ_0) is a point in the parametric envelope for (A,C) such that $(1-\theta_0/\theta^*)\rho_{\min}\leq\rho_0\leq(1-\theta_0/\theta^*)\rho_{\max}$ and $\theta_0\leq\theta^*$, then they show that $g(a,c|\rho_0,\theta_0)$ is a unique probabilistic mixture for (A,C).

Now, consider the following condition which Peterson and Reilly [23] and Reilly [27] refer to as a mixing condition :

$$g_{\min}(a_{i*},c_{j*})=g_{\max}(a_{i*},c_{j*})=0.\qquad(3)$$

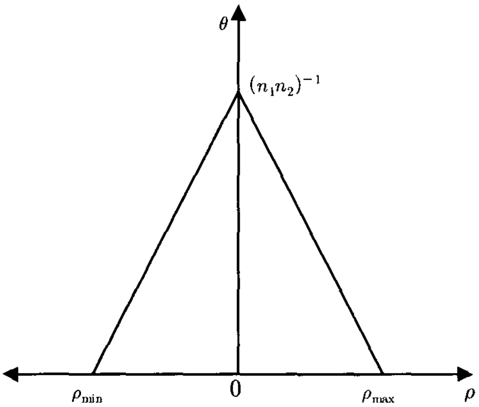
They show that if the mixing condition (3) is satisfied, then $\theta=\theta_0$ for pmf (2). Regardless of whether (3) is satisfied, $\rho=\rho_0$ for pmf (2).

Peterson and Reilly [23] call the pmf (2) a parametric mixture. In our computational experiment, sampling with explicitly induced correlation based on parametric mixtures is used to generate 0-1 knapsack problems.

3. Assumptions

In this study, we assume that A and C are discrete random variables such that $A\sim U\{j_1+1, j_1+2,\cdots,j_1+n_1\}$ and $C\sim U\{j_2+1,j_2+2,\cdots,j_2+n_2\}$. It follows that $\mu_A=j_1+(n_1+1)/2$, $\sigma_A^2=(n_1^2-1)/12$, $\mu_C=j_2+(n_2+1)/2$, and $\sigma_C^2=(n_2^2-1)/12$. Peterson and Reilly [23] show that when A and C are uniformly distributed, the parametric envelope is an isosceles triangle symmetric about $\rho=0$. The three points on the corners are $(0,(n_1n_2)^{-1})$, $(\rho_{\max},0)$, and $(\rho_{\min},0)$. Let $\theta^*=(n_1n_2)^{-1}$. See [Figure

1] for a representation of the parametric envelope for (A, C).



[Figure 1] Parametric envelope for (A, C)

We also assume that $n_2 \geq n_1 \geq 3$ and $m = n_2/n_1$ is integer. In this case, Relly [25, 26] and Peterson and Reilly [23] characterize the minimum-and maximum-correlation pmfs for (A, C) as follows :

$$g_{\min}(a,c) = \begin{cases} \frac{1}{n^2}, & if \ \ m(n_1+a-j_1) < (c-j_2) \\ & \quad \leq m(n_1+a-j_1+1) \quad ; \\ 0 & otherwise \ ; \end{cases}$$

and

$$g_{\max}(a,c) = \begin{cases} \frac{1}{n^2}, & if \ \ m(a-j_1-1) < (c-j_2) \\ & \quad \leq m(a-j_1) \\ 0 & otherwise \end{cases}$$

Furthermore,

$$\rho_{\max} = m((n_1^2-1)/(n_2^2-1))^{\frac{1}{2}}$$

and $\rho_{\min} = -\rho_{\max}$. The mixing condition (3) is satisfied.

For any point (ρ_0, θ_0) in the parametric envelope, there is a unique pmf that is mixture of $f_1(a)f_2(c)$, $g_{\min}(a,c)$, and $g_{\max}(a,c)$ [23, 25-27]. With our assumptions,

$$\lambda_0 = \theta_0/\theta^* = n_1 n_2 \theta_0 \ , \ \lambda_{\min} = (1 - n_1 n_2 \theta_0 - (\rho_0/\rho_{\max}))/2,$$

and

$$\lambda_{\max} = (1 - n_1 n_2 \theta_0 + (\rho_0/\rho_{\max}))/2$$

for parametric mixtures (2). With (2), λ_0, $\lambda_{\min}$, and $\lambda_{\max}$ are used to generate test problems for 0-1 knapsack problem.

4. Test Problem Description

In this study, we solve 2000 100-variable randomly generated 0-1 knapsack problems. The 0-1 knapsack problem has the following form :

Maximize

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

subject to

$$\sum_{j=1}^{n} a_j x_j \leq b$$

$$x_j = 0 \ or \ 1, \ j = 1, 2, \cdots, n,$$

where

$$x_j = \begin{cases} 1 & if \ item \ j \ is \ included \ in \ the \ knapsack \ ; \\ 0 & otherwise \ ; \end{cases}$$

$c_j > 0$ is the value of item j ; $a_j > 0$ is the weight of item j ; b is the capacity of the knapsack ; and n is the number of items to be considered for inclusion in the knapsack.

The 0-1 knapsack problem is known to be NP-complete. It is also known that the correlation between constraint coefficients and objective function coefficients has a strong effect on the number of iterations required to solve it with an implicit enumeration (branch and bound) routine.

We select 25 points from the parametric envelope, i.e., 25 different target combinations of ρ and θ (see [Figure 2]). The points are evenly separated horizontally as well as vertically. We select nine points from the bottom of the triangle and one point from the top of the triangle.

Between the top and the bottom, we select the number of points which is proportional to the width of the triangle at the specific θ value. These points are chosen to represent possible combinations of ρ and θ. We generate 80 100-variable knapsack problems to represent each point by using sampling with explicitly induced correlation based on parametric mixtures.

In order to see whether different combinations of m, n_1, n_2, j_1, and j_2 have an impact on the number of iterations to solve test problems with a branch and bound routine, we consider 16 different combinations of m, n_1, n_2, j_1, and j_2. We summarize those 16 combinations in <Table 1>. For each combination, 125 100-variable 0-1 knapsack problems are generated.

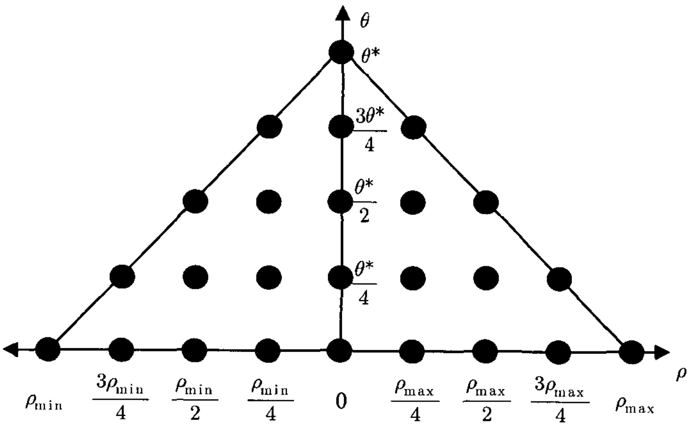
As a result, each set of 80 problems, which represents 25 different target combinations of ρ and θ, has 16 different combinations of m, n_1, n_2, j_1, and j_2. Hence, five problems in each set of 80 problems have a unique combination of m, n_1, n_2, j_1, and j_2. Similarly, each set of 125 problems, which represents 16 different combinations of m, n_1, n_2, j_1, and j_2, has 25 different target combinations of ρ and θ. Hence, five problems in each set of 80 problems have a unique target combination of ρ and θ.

In all of our test problems, the constant b, i.e., the right hand side of the constraint, is calculated as follows :

$$b = \lceil \frac{1}{2}\sum_{j=1}^{n} a_j \rceil .$$

<Table 1> All combinations of m, n_1, n_2, j_1, and j_2

Case	m	n_1	n_2	j_1	j_2
1	10	100	1000	0	0
2	10	100	1000	100	1000
3	10	100	1000	100	0
4	10	100	1000	0	1000
5	10	10	100	0	0
6	10	10	100	10	100
7	10	10	100	10	0
8	10	10	100	0	100
9	1	1000	1000	0	0
10	1	1000	1000	1000	1000
11	1	1000	1000	1000	0
12	1	1000	1000	0	1000
13	1	100	100	0	0
14	1	100	100	100	100
15	1	100	100	100	0
16	1	100	100	0	100



[Figure 2] 25 points from the parametric envelope

5. Generating and Solving Test Problems

In this section, we describe the algorithm GENER8 which is used to generate test problems and the multi-phase solution method that is used to solve those problems.

Our solution method is chosen based on availability and general acceptance of the procedures. They are a primal heuristic from Nemhauser and Wolsey [21] and CPLEX from ILOG, Inc.. CPLEX is contained in many commercially available packages and is available as a stand-alone product. Specifically, the mixed-integer optimizer in CPLEX, version 8.0 is used for our computational experiment. All of the necessary programs are coded in C, and the program is available upon request.

5.1 Generating Test Problems

The following algorithm GENER8 is based on a similar procedure in Reilly [27] where he assumes, $j_1 = 0$ and $j_2 = 0$.

Procedure GENER8

```
1. Generate $u_1, u_2, u_3 \sim U(0,1)$
2. $a \leftarrow \lfloor n_1 u_1 \rfloor + j_1 + 1$
3. If $u_3 \le \lambda_0$ $c \leftarrow \lfloor n_2 u_2 \rfloor + j_2 + 1$, and go to Step 6.
4. If $u_3 \ge 1 - \lambda_{max}$, $c \leftarrow \lfloor n_2 u_1 \rfloor + j_2 + 1$, and go to Step 6.
5. $c \leftarrow \lfloor n_2 (1 - u_1) \rfloor + j_2 + 1$.
6. Return $(a,c)$.
```

A random number seed, 98765, is used to get pseudo random numbers for all 16 combinations of m, n_1, n_2, j_1, and j_2. The GENER8 repeats 100 times to generate one test problem with a combination of m, n_1, n_2, j_1, and j_2 since the number of variable is 100. A detailed C++ code is available upon request.

5.2 Solving Test Problems

The decision variables in each test problem are sorted so that $c_i/a_i \ge c_{i+1}/a_{i+1}$ for $i = 1, 2, \cdots, 99$.

5.2.1 Primal Heuristic

After the sorting procedure is executed and the LP (Linear Programming) relaxation is solved, a primal greedy heuristic ([21], p. 452) is applied to the test problem.

Let N^1 and N^0 denote sets of the variable x_j whose values are 1 and 0, respectively, in the solution to the LP relaxation. Then, $x_j = 1$ for $j \in N^1 = \{1, 2, \cdots, r-1\}$, $x_j = 0$ for $j \in N^0 = \{r+1, r+2, \cdots, n\}$, and $x_r = (b - \Sigma_{j \in N^1} a_j / a_r)$ is the solution to the LP relaxation for $r \in \{1, 2, \cdots, n\}$. Let $\hat{x}_j$, $j = 1, 2, \cdots, n$, represent the heuristic solution. The following is the statement of the primal heuristic [21] :

Primal Heuristic

```
1. Set $\hat{x}_j = 1$ for all $j \in N^1$, and $\hat{x}_r = 0$.
2. Set $b \leftarrow b - \Sigma_{j \in N^1} a_j$ and $j = r + 1$.
3. If $a_j > b$, set $\hat{x}_j = 0$; otherwise, set $\hat{x}_j = 1$ and
   $b \leftarrow b - a_j$.
4. If $j < n$, $j \leftarrow j + 1$ and go to Step 3 ; otherwise,
   stop.
```

Basically, the primal heuristic above uses the following idea. By solving the LP relaxation, we can identify critical index, r. After subtracting all constraint coefficients whose indices are less than r from the original right hand side, we set the index of x_j at $r+1$ and try to fix x_j to 1 by subtracting the coefficient of x_j from the remainder of the right hand side. We continue to increase the index until the updated right hand side would become negative if any more variables were fixed at 1.

In Section 5, we examine the relative errors of this heuristic by comparing the optimal values with the heuristic values.

5.2.2 A Representative Algorithm : CPLEX

The mixed integer optimizer in CPLEX, version 8.0, is used to solve the test problems. The number of nodes is used in this study to measure CPLEX performance. Note that Hill and Reilly [7] and Cario et al. [3] also used the number of CPLEX nodes to measure the hardness of their test problems.

6. Computational Results and Analysis

In this section, we summarize and analyze our results. In the first subsection, we consider three indicators of problem hardness, (1) the population correlation (ρ) between objective function and constraint coefficients, (2) the sample correlation, and (3) the smallest joint probability (θ) for any possible coefficient combination. In the next section, we compare the results with different combinations of m, n_1, n_2, j_1, and j_2. Finally, we examine the relative errors of the primal heuristic in the third subsection.

6.1 Three Indicators of Problem Hardness

We consider three problem hardness indicators or independent variables in this subsection. They are the population correlation (ρ) between objective function and constraint coefficients, the sample correlation, and the smallest joint probability (θ) for any possible coefficient combination. The number of CPLEX nodes is the dependent variable in this subsection.

Because it takes much less than a second to calculate these indicators or they are known in advance, if they predict the problem hardness of the test problems well, we can estimate whether any problem is easy or hard to solve and perhaps choose a solution method or heuristic accordingly. By understanding the relationship between θ and ρ and problem hardness, we may be able to generate test problems that are likely to be difficult to solve.

Determining each indicator for the test problems is much easier than solving those problems. If we can find a good indicator, then we can generate test problems which can be characterized by that indicator. In other words, we may generate synthetic discrete optimization test problems with desired hardness properties.

6.1.1 Correlation, ρ

Each cell in <Table 2> represents a target combination of ρ and θ from parametric envelope. The value in each cell is obtained by taking the average of the number of nodes over 80 test problems and it is presented with associated standard error. The data for numbers of nodes are from running CPLEX optimization. Every single problem is successfully solved by CPLEX. The maximum number of nodes is 165093832, which is obtained when $\rho=\rho_{\max}$ and $\theta=0$.

As expected, the average number of nodes clearly increase as the target correlations increase, and the maximum number of nodes occurs when $\rho=\rho_{\max}$. This trend is also clear for each $\theta \in \{0,\theta^*/4,\theta^*/2,3\theta^*/4,\theta^*\}$. Hence, we can conclude that with larger target correlations, we have better chance to generate harder test problems for CPLEX.

6.1.2 Sample Correlation

The values of average sample correlation in <Table 3> are nearly the same as the corresponding target population correlations, even though there is some sampling error evident in each cell. This confirms that our generation procedure for testing procedure works well for the 0-1 knapsack problem.

In [Figure 3], there is a plot of sample correlations vs. Log (number of iterations). We can see a clear trend that many hard problems are associated with large sample correlation, and especially 'very hard problems' are associated with correlations close to 1.

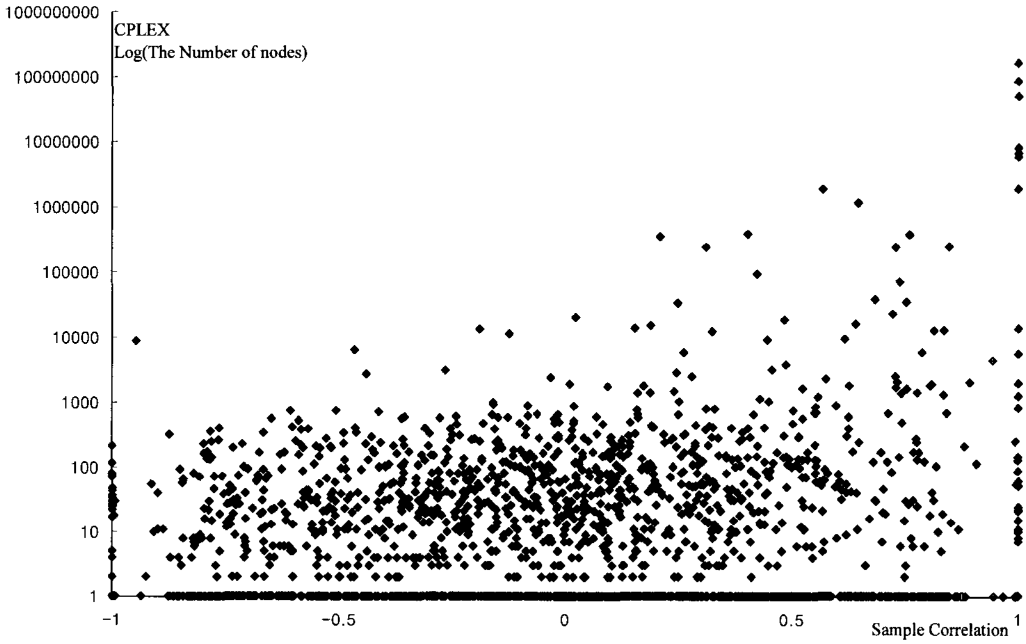
Hence, we can affirm the idea about the effect of the correlation between A and C on the performance of CPLEX routine. The number of nodes seems to increase exponentially as the sample correlation between A and C increases.

〈Table 2〉 The average number of nodes (standard error)

	$\rho_{\min}$	$\frac{3\rho_{\min}}{4}$	$\frac{\rho_{\min}}{2}$	$\frac{\rho_{\min}}{4}$	0	$\frac{\rho_{\max}}{4}$	$\frac{\rho_{\max}}{2}$	$\frac{3\rho_{\max}}{4}$	$\rho_{\max}$	Avg.
θ^*					39 (2.24)					39
$\frac{3\theta^*}{4}$				29 (1.52)	78 (4.67)	74 (6.54)				60
$\frac{\theta^*}{2}$			38 (2.01)	66 (3.93)	68 (6.23)	106 (6.34)	112 (6.99)			78
$\frac{\theta^*}{4}$		45 (2.99)	57 (4.14)	69 (4.23)	90 (8.70)	133 (12.32)	286 (34.00)	5281 (860.61)		852
0	12 (0.92)	135 (29.25)	62 (3.73)	492 (62.75)	4740 (1159.7)	4215 (807.9)	30765 (6442.9)	22427 (4111.1)	4069407 (636374)	459139
Avg.	12	90	52	164	1003	1132	10388	13854	4069407	165553

〈Table 3〉 Sample correlation (standard error)

	$\rho_{\min}$	$\frac{3\rho_{\min}}{4}$	$\frac{\rho_{\min}}{2}$	$\frac{\rho_{\min}}{4}$	0	$\frac{\rho_{\max}}{4}$	$\frac{\rho_{\max}}{2}$	$\frac{3\rho_{\max}}{4}$	$\rho_{\max}$
θ^*					0.011 (0.0029)				
$\frac{3\theta^*}{4}$				0.244 (0.0032)	0.009 (0.0033)	0.240 (0.0034)			
$\frac{\theta^*}{2}$			0.474 (0.0032)	0.248 (0.0032)	0.020 (0.0035)	0.258 (0.0037)	0.503 (0.0027)		
$\frac{\theta^*}{4}$		0.763 (0.0024)	0.525 (0.0030)	0.260 (0.0033)	0.016 (0.0036)	0.235 (0.0036)	0.500 (0.0028)	0.757 (0.0020)	
0	0.999 (0.0001)	0.750 (0.0026)	0.534 (0.0031)	0.245 (0.0035)	0.007 (0.0037)	0.236 (0.0042)	0.491 (0.0032)	0.766 (0.0023)	0.999 (0.0001)



[Figure 3] Sample correlation vs. log (the number of nodes)

6.1.3 Minimum Joint Probability, θ

The smallest joint probability for any possible combination of coefficients, θ, is the third indicator of problem hardness. From <Table 2>, we can clearly see that the smaller value of θ leads to a greater average number of CPLEX nodes. This trend can be seen more clearly if we fix target correlation and compare the numbers of nodes in the same column. As a result, we conclude that the number of iterations increases as θ decreases.

6.2 Comparing Result with Different Combinations of m, n_1, n_2, j_1, and j_2.

We set up 16 different combinations of coefficient range parameters to look at the effect of each parameter m, n_1, n_2, j_1, and j_2 on the number of nodes to solve test problems.

<Table 4> summarizes the average number of CPLEX nodes for different n_1 and n_2 combinations and the number of nodes in each cell is calculated by taking the average over 500 problems. The result shows that given m value, problems with larger n_1 and n_2 values are harder than those with smaller n_1 and n_2.

Also, we can observe that the test problems with larger $\min\{n_1, n_2\}$ seem to be harder than those with smaller $\min\{n_1, n_2\}$.

<Table 4> The average number of nodes for combinations of n_1 and n_2

(n_1, n_2)			
(100,1000)	(10,100)	(1000,1000)	(100,100)
26996	15.25	634880.3	326

We summarize the average number of nodes for different combinations of m, n_1, n_2, j_1, and j_2

in <Table 5>.

The average number of nodes is 13506 when $m=10$ and 317603 when $m=1$. So, the problems generated with $m=1$ are more difficult to solve than those generated with $m=10$. This trend is clearer if we compare Case 1 with Case 8, Case 2 with Case 9, Case 3 with Case 10 and etc. in <Table 5>.

The results also indicate that when we have larger n_1 and n_2, the test problems for which either $j_1=0$ or $j_2=0$ are more difficult than the problems for which both $j_1=0$ and $j_2=0$ or both $j_1>0$ and $j_2>0$. Moreover, when we have larger n_1 and n_2, the test problems for which $j_1=0$ only are more difficult than the problems for which $j_2=0$ only.

<Table 5> The number of nodes (standard error) for combinations of m, n_1, n_2, j_1 and j_2.

Case	Number of nodes (standard error)	m	n_1	n_2	j_1	j_2
1	23 (2.01)	10	100	1000	0	0
2	155 (8.68)	10	100	1000	100	1000
3	616 (81.31)	10	100	1000	100	0
4	107190 (29901.46)	10	100	1000	0	1000
5	1 (0.13)	10	10	100	0	0
6	26 (2.37)	10	10	100	10	100
7	23 (2.20)	10	10	100	10	0
8	11 (1.20)	10	10	100	0	100
9	19 (1.76)	1	1000	1000	0	0
10	116 (8.70)	1	1000	1000	1000	1000
11	35708 (7676.95)	1	1000	1000	1000	0
12	2503657 (640403.19)	1	1000	1000	0	1000
13	10 (0.95)	1	100	100	0	0
14	212 (30.23)	1	100	100	100	100
15	529 (133.73)	1	100	100	100	0
16	553 (134.19)	1	100	100	0	100

After observing <Table 5>, we may conclude that both n_1, n_2 and j_1, j_2 play roles to determine the hardness of test problems.

6.3 Relative Error of the Primal Heuristic Algorithm

Calculating the relative error for the heuristic is worthwhile to do because we can see how the heuristic works with different combinations of θ and ρ. Let the optimal value of the test problem be z^*, the heuristic value of the test problems be $\hat{z}$, and the relative error be

$$\frac{z^* - \hat{z}}{z^*}.$$

From <Table 6> and <Table 7>, we can not observe any clear trend in the relative error; all numbers are around 0.002~0.004 with standard error of about 0.0001. We highly suspect that the number of variables in our problem is relatively too big to see differences in relative errors. In other words, making a couple of bad selections has relatively small impact on the optimal value of 0-1 knapsack problem when the problem size is large compared to a small problem. To support this conjecture, we solve additional 2000 10-variable 0-1 knapsack problems with the same assumption we use for the 2000 100-varible problems. The result is presented in <Table 8> and <Table 9> in Appendix A. We can see that the average relative errors increase as ρ increases in <Table 8>. In <Table 9>, the results indicate that when we have $n_1>0$, $n_2>0$, $j_1>0$ and $j_2>0$, the test problems seem to be more difficult than other cases. Also, with the same n_1 and n_2, the test problems with either $j_1=0$ and $j_2>0$ are more difficult than the problems with both $j_1>0$ and $j_2=0$. Moreover, when we have both $j_1=0$

〈Table 6〉 The average relative errors (standard error) when $n=100$

	$\rho_{\min}$	$\frac{3\rho_{\min}}{4}$	$\frac{\rho_{\min}}{2}$	$\frac{\rho_{\min}}{4}$	0	$\frac{\rho_{\max}}{4}$	$\frac{\rho_{\max}}{2}$	$\frac{3\rho_{\max}}{4}$	$\rho_{\max}$
θ^*					0.0025 (0.0001)				
$\frac{3\theta^*}{4}$				0.0028 (0.0001)	0.0033 (0.0001)	0.0030 (0.0001)			
$\frac{\theta^*}{2}$			0.0022 (0.0001)	0.0029 (0.0001)	0.0032 (0.0001)	0.0032 (0.0001)	0.0024 (0.0001)		
$\frac{\theta^*}{4}$		0.0020 (0.0001)	0.0029 (0.0001)	0.0032 (0.0001)	0.0030 (0.0001)	0.0033 (0.0001)	0.0031 (0.0001)	0.0032 (0.0001)	
0	0.0000 (0.0000)	0.0027 (0.0001)	0.0105 (0.0020)	0.0026 (0.0001)	0.0031 (0.0001)	0.0028 (0.0001)	0.0025 (0.0001)	0.0029 (0.0001)	0.0025 (0.0001)

〈Table 7〉 The average relative errors (standard error) for combinations of m, n_1, n_2, j_1 and j_2 when $n=100$.

Case	Number of nodes (standard error)	m	n_1	n_2	j_1	j_2
1	0.0197 (0.0011)	10	100	1000	0	0
2	0.0292 (0.0011)	10	100	1000	100	1000
3	0.0040 (0.0005)	10	100	1000	100	0
4	0.1973 (0.0009)	10	100	1000	0	1000
5	0.0099 (0.0007)	10	10	100	0	0
6	0.0274 (0.0012)	10	10	100	10	100
7	0.0105 (0.0021)	10	10	100	10	0
8	0.0178 (0.0009)	10	10	100	0	100
9	0.0256 (0.0015)	1	1000	1000	0	0
10	0.0374 (0.0013)	1	1000	1000	1000	1000
11	0.0086 (0.0007)	1	1000	1000	1000	0
12	0.0220 (0.0010)	1	1000	1000	0	1000
13	0.0251 (0.0014)	1	100	100	0	0
14	0.0292 (0.0012)	1	100	100	100	100
15	0.0069 (0.0006)	1	100	100	100	0
16	0.0226 (0.0010)	1	100	100	0	100

and $j_2=0$, the test problems with $m=1$ seem to be more difficult than the problems with $m=10$.

The result also may be due to the simple structure of the knapsack problem because even the simple heuristic can provide relatively good approximation to the optimal solution as the number of variables increases. Hence, we may conclude that the performance of the heuristic in terms of relative errors does not have any clear relationship with the parameters we controlled when test problems are relatively large.

7. Discussions

The computational experiment was performed to determine the effects of different parameters such as θ, ρ, m, n_1, n_2,j_1, and j_2 for 0-1 knapsack problem on the performance of an exact and a heuristic solution procedure for the 0-1 knapsack problem. The test problems were generated with explicitly induced correlation based on parametric mixtures.

The results for 2000 100-variable test problems generated indicate that performance of CPLEX solution procedures generally degrades as the target correlation between the objective function and the constraint coefficients increases from $\rho_{\min}$ to $\rho_{\max}$. The performance of the CPLEX solution performance also degrades as the target θ decreases from θ^* to 0.

We set up 16 different combinations of coefficient range parameters to look at the effect of each parameter m, n_1, n_2, j_1, and j_2 on the number of nodes to solve test problems. The result shows that given m value, problems with larger n_1 and n_2 values are harder than those with smaller n_1 and n_2. Also, we can observe that the test problems with larger $\min\{n_1,n_2\}$ seems to be harder than those with smaller $\min\{n_1,n_2\}$.

The result indicates that problems generated with $m=1$ are more difficult to solve than those generated with $m=10$. Moreover, different combinations of j_1 and j_2 also lead to different hardness of the test problems.

We could not find any clear trend among relative errors for the heuristic but we may conjecture that if the problem size is reasonably large, the relative error from the primal heuristic is small regardless of the combinations of θ and ρ. To support this idea, we solve additional 2000 10-variable 0-1 knapsack problem with the same assumption we use for the 2000 100-varible problems. The result is presented in <Table 8> and <Table 9> in Appendix A.

When conducting computational experiments, we recommend that synthetic problems be generated carefully by using the method such as generation with the explicitly induced correlation. More challenging problems can be generated with this type of generation and can lead to a better understanding of the capabilities and limitations of the solution method(s) begin evaluated. Additionally, the experiment will provide a clearer indication of the effect of correlations [3]. Several researches provide recommendations for computational experiment including performing careful pilot studies to identify the most crucial factors and treatment levels for the subsequent experiment [2, 3, 7].

For the 0-1 knapsack problems, we recommend any future computational experiment includes different combinations of θ, ρ, m, n_1, n_2, j_1, and j_2 as we performed. Also, authors may try different right hand side levels and different number of variables to cover some of the unused factors in our experiment.

We admit that generalization of the result in this paper has a few important limitations. There are many different algorithms available for 0-1 knapsack problems including heuristics. For some algorithms, correlation structure of the problem is not related to performance of the algorithm. For example, a typical dynamic programming (DP) algorithm for 0-1 knapsack problems runs in $O(nb)$ time, and it is clear that the best indicators of problem hardness when we use DP algorithm are the number of variables and right hand side value. Hence, when the run time for an algorithm can be easily recognized in terms of given parameters, we do not recommend the approach which is used in the paper.

Another limitation may be the types of problems which we can apply our result to. Some problems may have just one type of coefficients so that it may be impossible to apply our approach. For the case of a two machine parallel shop scheduling problem with the objective of minimizing makespan, the objective function can be just one variable with a coefficient value of one and all coefficients of constraints can be 1 or -1. Other real world problems may be too complicated to apply our approach because there exist several different types of constraints with different coefficient values. Hence, care must be taken when the result is applied to other types of problems.

References

[1] Balas, E. and E. Zemeal, "An Algorithm for Large Zero-One Knapsack Problems," *Operations Research*, Vol.28, No.4(1980), pp.1130-1154.

[2] Barr, R.S., B.L. Golden, J.P. Kelly, M.G.C. Resende, and W.R. Stewart Jr., "Designing and Reporting on Computational Experiments with Heuristic Methods," *Journal of Heuristics*, Vol.1, No.1(1995), pp.9-32.

[3] Cario, M.C., J.J. Clifford, R.R. Hill, J. Yang, K. Yang, and C.H. Reilly, "An Investigation of the Relationship between Problem Characteristics and Algorithm Performance : a Case Study of the GAP," *IIE Transactions*, Vol.34, No.3(2002), pp.297-312.

[4] Fayard, D. and G. Plateau, "An Algorithm for the Solution of the 0-1 Knapsack Problem," *Computing*, Vol.28(1982), pp.269-287.

[5] Hooker, J.N., "Needed : an Empirical Science of Algorithms," *Operations Research*, Vol.42, No.2(1994), pp.201-212.

[6] Hooker, J.N., "Testing Heuristics; We Have It All Wrong," *Journal of Heuristics*, Vol.1, No.1(1995), pp.33-32.

[7] Hill, R.R. and C.H. Reilly, "The Effect of Coefficient Correlation Structure in Two-Dimensional Knapsack Problems on Solution Procedure Performance," *Operations Research*, Vol.46, No.2(2000), pp.302-317.

[8] Martello, S. and P. Toth, "Algorithms for the Solution of the 0-1 Single Knapsack Problem," *Computing*, Vol.21(1978), pp.81-86.

[9] Martello, S. and P. Toth, "Algorithms for Knapsack Problems," Surveys in Combinatorial Optimization, pp.213-257, Elsevier Science Publishers B.V., Amsterdam, Netherlands, 1979.

[10] Martello, S. and P. Toth, "The 0-1 Knapsack Problem," Combinatorial Optimization, eds. N. Christofides, A. Mingozzi, C. Sandi, pp.237-279, John Wiley and Sons, New York, New York, 1979.

[11] Martello, S. and P. Toth, "A New Algorithm for the 0-1 Knapsack Problem," *Management Science*, Vol.35, No.5(1988), pp.633-644.

[12] Martello, S. and P. Toth, "An Exact Algorithm for Large Unbounded Knapsack Problems," *Operations Research Letters*, Vol.35, No.9(1990), pp.15-20.

[13] Martello, S., D. Pisinger, and P. Toth, "Dynamic Programming and Strong Bounds for the 0-1 Knapsack Problem," *Management Science*, Vol.45, No.3(1999), pp.414-424.

[14] Martello, S., D. Pisinger, and P. Toth, "New trends in exact algorithms for the 0-1 Knapsack Problem," *European Journal of Operational Research*, Vol.123(2000), pp.325-332.

[15] Moore, B.A., "Correlated 0-1 Knapsack Problems," IND ENG 854 Course Project, Department of Industrial and Systems Engineering, The Ohio State University, Columbus, Ohio, 1989.

[16] Moore, B.A., "The Effect of Correlation on Exact and Heuristic Procedures for the Weighted Set Covering Problem," M.S. Thesis, Department of Industrial and Systems Engineering, The Ohio State University, Columbus, Ohio, 1990.

[17] Moore, B.A., J.A. Peterson, and C.H. Reilly, "Characterizing Distributions of Discrete Bivariate Random Variables for Simulation and Evaluation of Solution Methods," *Pro-*

ceedings of the 1990 Winter Simulation Conference, eds. O. Baci, R.P. Sadowski, R.E. Nance, pp.294-302, Institute of Electrical and Electronics Engineers, New Orleans, Louisiana, 1990.

[18] Moore, B.A. and C.H. Reilly, "Randomly Generating Optimization Test Problems with Controlled Correlation," Working Paper 1992-001, Department of Industrial and Systems Engineering, The Ohio State University, Columbus, Ohio, 1992.

[19] Moore, B.A. and C.H. Reilly, "Randomly Generating Synthetic Optimization Problems with Explicitly Induced Correlation," Working Paper 1993-002, Department of Industrial and Systems Engineering, The Ohio State University, Columbus, Ohio, 1993.

[20] Nauss, R.M., "An Efficient Algorithm for the 0-1 Knapsack Problem," *Management Science*, Vol.23, No.1(1976), pp.27-31.

[21] Nemhauser, G.L. and L.A. Wolsey, *Integer and Combinatorial Optimization*, Wiley and Sons, New York, New York, 1988.

[22] Peterson, J.A., "A Parametric Analysis of a Bottleneck Transportation Problem Applied to the Characterization of Correlated Discrete Bivariate Random Variables," M.S. Thesis, Department of Industrial and Systems Engineering, The Ohio State University, Columbus, Ohio, 1990.

[23] Peterson, J.A. and C.H. Reilly, "Joint Probability Mass Functions for Coefficients in Synthetic Optimization Problems," Working Paper 1993-006, Department of Industrial and Systems Engineering, The Ohio State University, Columbus, Ohio, 1993.

[24] Pisinger, D., "A minimal algorithm for the 0-1 Knapsack Problem," *Operations Research*, Vol.45, No.5(1997), pp.758-767.

[25] Reilly, C.H., "Optimization Test Problems with Uniformly Distributed Coefficients," *Proceedings of the 1991 Winter Simulation Conference*, eds. B.L. Nelson, W.D. Kelton, G.M. Clark, pp.866-874, Institute of Electrical and Electronics Engineers, Phoenix, Arizona, 1991.

[26] Reilly, C.H., "Comparison of Alternative Input Models for Synthetic Optimization Problems," *Proceedings of the 1993 Winter Simulation Conference*, eds. G.W. Evans, M. Mollaghasemi, E.C. Russel, W.E. Biles, pp. 356-364, Institute of Electrical and Electronics Engineers, Los Angeles, California, 1993.

[27] Reilly, C.H., "Alternative Input Models for Generating Synthetic Optimization Problems : Analysis and Implication," Working Paper 1994-001, Department of Industrial and Systems Engineering, The Ohio State University, Columbus, Ohio, 1994.

[28] Reilly, C.H., "Optimization Test Problems With Uniformly Distributed Coefficients," *Proceedings of the 1999 Winter Simulation Conference*, eds. P.A. Farrington, H.B. Nembhard, D.T. Strurrock, and G.E. Evans, pp.116-121, Institute of Electrical and Electronics Engineers, Phoenix, Arizona, 1999.

[29] Rushmeier, R.A. and G.L. Nemhauser, "Experiments with Parallel Branch-and-Bound Algorithms for the Set Covering Problem," *Operations Research Letters*, Vol.13, No.5 (1993), pp.277-285.

[30] Sahni, S., "Approximate Algorithms for the 0/1 Knapsack Problem," *Journal of the Association for Computing Machinery*, Vol. 22, No.1(1975), pp.115-124.

Appendix A.

〈Table 8〉 The average relative errors (standard error) when $n=10$.

	$\rho_{\min}$	$\frac{3\rho_{\min}}{4}$	$\frac{\rho_{\min}}{2}$	$\frac{\rho_{\min}}{4}$	0	$\frac{\rho_{\max}}{4}$	$\frac{\rho_{\max}}{2}$	$\frac{3\rho_{\max}}{4}$	$\rho_{\max}$
θ^*					0.0161 (0.0008)				
$\frac{3\theta^*}{4}$				0.0159 (0.0007)	0.0231 (0.0009)	0.0238 (0.0009)			
$\frac{\theta^*}{2}$			0.0114 (0.0006)	0.0128 (0.0007)	0.0188 (0.0008)	0.0260 (0.0010)	0.0296 (0.0011)		
$\frac{\theta^*}{4}$		0.0060 (0.0004)	0.0106 (0.0006)	0.0144 (0.0007)	0.0255 (0.0009)	0.0253 (0.0010)	0.0278 (0.0009)	0.0284 (0.0010)	
0	0.0000 (0.0000)	0.0084 (0.0005)	0.0104 (0.0005)	0.0184 (0.0008)	0.0248 (0.0009)	0.0253 (0.0010)	0.0298 (0.0011)	0.0313 (0.0021)	0.0295 (0.0011)

〈Table 9〉 The average relative errors (standard error) for combinations of m, n_1, n_2, j_1 and j_2 when $n=10$.

Case	Number of nodes (standard error)	m	n_1	n_2	j_1	j_2
1	0.0197 (0.0011)	10	100	1000	0	0
2	0.0292 (0.0011)	10	100	1000	100	1000
3	0.0040 (0.0005)	10	100	1000	100	0
4	0.0197 (0.0009)	10	100	1000	0	1000
5	0.0099 (0.0007)	10	10	100	0	0
6	0.0274 (0.0012)	10	10	100	10	100
7	0.0105 (0.0021)	10	10	100	10	0
8	0.0178 (0.0009)	10	10	100	0	100
9	0.0256 (0.0015)	1	1000	1000	0	0
10	0.0374 (0.0013)	1	1000	1000	1000	1000
11	0.0086 (0.0007)	1	1000	1000	1000	0
12	0.0221 (0.0010)	1	1000	1000	0	1000
13	0.0251 (0.0014)	1	100	100	0	0
14	0.0292 (0.0012)	1	100	100	100	100
15	0.0069 (0.0006)	1	100	100	100	0
16	0.0226 (0.0010)	1	100	100	0	100