3-1-2005

Developing New Multidimensional Knapsack Heuristics Based on Empirical Analysis of Legacy Heuristics

Yong Kun Cho

DEVELOPING NEW MULTIDIMENSIONAL KNAPSACK HEURISTICS BASED ON EMPIRICAL ANALYSIS OF LEGACY HEURISTICS

DISSERTATION

Yong Kun Cho, Major, Republic of Korea Army

AFIT/DS/ENS/05-01

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT/DS/ENS/05-01

DEVELOPING NEW MULTIDIMENSIONAL KNAPSACK HEURISTICS BASED ON EMPIRICAL ANALYSIS OF LEGACY HEURISTICS

DISSERTATION

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

Yong Kun Cho, B.S., M.S.

Major, Republic of Korea Army

March 2005

AFIT/DS/ENS/05-01

DEVELOPING NEW MULTIDIMENSIONAL KNAPSACK HEURISTICS BASED ON EMPIRICAL ANALYSIS OF LEGACY HEURISTICS

Yong Kun Cho, B.S., M.S.
Major, Republic of Korea Army

Approved:

/signed/

____________________________________ ____________

Dr. James T. Moore (Co-Chairman) date

/signed/

____________________________________ ____________

Dr. Raymond R. Hill (Co-Chairman) date

/signed/

____________________________________ ____________

Dr. Timothy S. Webb (Member) date

/signed/

____________________________________ ____________

Dr. Dennis W. Quinn (Dean's Representative) date

Accepted:

/signed/

____________________________________ ____________

Robert A. Calico, Jr. date

Dean, Graduate School of Engineering and Management

AFIT/DS/ENS/05-01

Abstract

The multidimensional knapsack problem (MKP) has been used to model a variety of practical optimization and decision-making applications. Due to its combinatorial nature, heuristics are often employed to quickly find good solutions to MKPs. While there have been a variety of heuristics proposed for the MKP, and a plethora of empirical studies comparing the performance of these heuristics, little has been done to garner a deeper understanding of heuristic performance as a function of problem structure. This dissertation presents a research methodology, empirical and theoretical results explicitly aimed at gaining a deeper understanding of heuristic procedural performance as a function of test problem characteristics. This work first employs an available, robust set of two-dimensional knapsack problems in an empirical study to garner performance insights. These performance insights are tested against a larger set of problems, five-dimensional knapsack problems specifically generated for empirical testing purposes. The performance insights are found to hold in the higher dimensions. These insights are used to formulate and test a suite of three new greedy heuristics for the MKP, each improving upon its successor. These heuristics are found to outperform available legacy heuristics across a complete spectrum of test problems. Problem reduction heuristics are examined and the subsequent performance insights garnered are used to derive a new problem reduction heuristic, which is then further extended to employ a local improvement phase. These problem reduction heuristics are also found to outperform currently available approaches. Available problem test sets are shown lacking along multiple dimensions of importance for viable empirical testing. A new problem generation methodology is developed and shown to overcome the current limitations in available problem test sets. This problem generation methodology is used to generate a new set of empirical test problems specifically designed for competitive computational tests. This new test set is shown to stress existing heuristics; not only does the computational time required by these legacy heuristics increase with problem size, but solution quality is found to decrease with problem size. However, the solution quality obtained by the suite of heuristics developed in this dissertation are shown to be unaffected by problem size thereby providing a level of robust solution quality not previously seen in heuristic development for the MKP. This research demonstrates that the test problems can have a profound, and sometimes misleading, impact on the general insights gained via empirical testing, provides six new quality heuristics, and two new robust sets of test problems, one focused on empirical testing, the other focused on competitive testing.

Acknowledgements

Upon completing this dissertation, I would like to express my deepest gratitude to my advisors, Dr. James T. Moore and Dr. Raymond R. Hill. Their constant support and leadership enabled me to put in my best efforts over the past three years in accomplishing this achievement. It was definitely a great honor and an invaluable experience to work with Dr. Moore, who gave continuous support until the last minute. Dr. Hill, whom I am indebted to since my master years, has shared his academic insights, encouraged my wide range research efforts and sincerely supported my work from beginning to end. I greatly respect and admire his professionalism and his kind words of encouragement will never be forgotten. Also, I appreciate my committee member, Dr. Timothy S. Webb, for his time and dedication.

Furthermore, I thank Korean Army for giving me the opportunity to study Doctoral degree program at Air Force Institute of Technology. Studying abroad in a military school has broadened my academic knowledge and stimulated my understanding for military technology. I hope this priceless experience helps in better serving my country in the future.

Finally, my great thanks go to my family. My irreplaceable wife and two cute sons deserve to have the greatest appreciation for their love, encouragement and sacrifices. I also feel grateful for my parents and in-law's in Korea for their love and support. Academic success, service to my country, and the love of my family are gifts for which I am eternally grateful.

Yong Kun Cho

Table of Contents

Page

Page

List of Figures

Figure Page

Figure Page

List of Tables

Table Page

Table Page

Table Page

DEVELOPING NEW MULTIDIMENSIONAL KNAPSACK HEURISTICS BASED ON EMPIRICAL ANALYSIS OF LEGACY HEURISTICS

I. Introduction

1.1 General Discussion

Obtaining an exact solution to an integer programming problem in real practice is sometimes less practical in comparison to an easily computed method of acquiring near-optimal solutions via heuristics. As problem formulations get large, finding exact solutions often requires excessive computing time and storage space. These large problem formulations often involve parameters that are abstractions of reality so an optimal solution may not have as much value as one would hope. Considering the imprecision of real-world problem data and that a precise solution in reality may not be desired because of the time and effort required to achieve it, obtaining a near-optimal solution in a reasonable amount of time may better satisfy a real world practitioner.

The knapsack problem (KP) is an integer programming problem with wide application in many areas. Its general form, the multidimensional knapsack problem (MKP), has frequently been used to model various decision-making processes such as manufacturing in-sourcing (Cherbaka *et al.*, 2004), asset-backed securitization (Mansini and Speranza, 2002), combinatorial auctions (DeVries and Vohra, 2000; Rothkopf *et al.*, 1995), computer systems design (Ferreira *et al.*, 1993), resource-allocation (Johnson *et al.*, 1985), set packing (Fox and Scudder, 1985), cargo loading (Shih, 1979), project selection (Peterson, 1967), cutting stock (Gilmore and Gomery, 1966), and capital budgeting (early examples include Lorie and Savage, 1955; Manne and Markowitz, 1957; Weingartner, 1966).

The multidimensional knapsack problem involves multiple resource constraints. The standard knapsack problem, with one constraint, is known to be NP-hard (Garey and Johnson, 1979), hence the 0 – 1 MKP is also an NP-hard problem (Frieze and Clarke, 1984). An efficient heuristic provides a good solution, not necessarily optimal, using a reasonable amount of time and resources. Heuristics are generally efficient in terms of solution quality, computer resource requirements and computer solving time. However, it should also be mentioned that, according to Magazine and Chern (1984), finding a polynomial-time approximation for the MKP is itself an NP-hard problem.

Many researchers have developed heuristic methods to solve MKPs. These heuristic approaches vary in how they treat the problem and how they select items for inclusion in the knapsack. Little, if anything, has been done to understand how heuristic procedural differences affect performance. This dissertation provides empirical analyses of heuristics for MKPs and examines how various heuristics function according to particular test problem characteristics. New heuristics are introduced based on the knowledge gained from the empirical analysis of MKPs.

1.2 Knapsack Problems

Suppose there are n items. The jth item, $j = 1,\ldots, n$, has a cost a_j and a value c_j. An item is either picked or rejected. There is a resource limit of b available for which the items compete. The problem of choosing a subset of items to maximize the sum of the values while not exceeding the resource constraint is the 0 – 1 knapsack problem,

$$\max\left\{\sum_{j=1}^{n} c_j\, x_j : \sum_{j=1}^{n} a_j\, x_j \le b,\ x_j = 0 \ \text{ or } \ 1, \ \ j = 1, ..., n\right\} \ . \tag{1}$$

This problem is called the knapsack problem because of the analogy to a hiker's problem of deciding what to put in a knapsack given a weight or volume limitation on the knapsack.

In general, problems of this sort may have *m* constraints. These problems are referred as the multidimensional knapsack problem (MKP) (Nemhauser and Wolsey, 1988). With the MKP, item selection must simultaneously satisfy all *m* constraints.

The MKP is of the following form:

Maximize

$$Z = \sum_{j=1}^{n} c_j\, x_j \tag{2}$$

subject to

$$\sum_{j=1}^{n} a_{ij}\, x_j \le b_i \qquad i = 1, 2, \ . \ . \ . \ , m \tag{3}$$

$$x_j = 0 \quad \text{or} \quad 1 \qquad j = 1, 2, \ . \ . \ . \ , n \tag{4}$$

where $c_j > 0$, $b_i > 0$ and all $a_{ij} \ge 0$. Additionally at least one $a_{ij} > 0$ for each *j*.

Another well-known knapsack problem is the multi-knapsack problem. The multi-knapsack problem deals with *m* distinct knapsacks with *n* given items. The *m* knapsacks have capacities b_i, $i = 1, 2, \ldots, m$. Each item has a profit c_{ij} and a cost a_{ij} associated with each knapsack. The problem is to choose *m* disjoint subsets from the *n* items, such that the sum of the value of the selected items is maximized while the cost of the selected items do not exceed the capacity of any knapsack *i*, for each $i \in \{1, 2, ..., m\}$. The multi-knapsack problem can be formulated as follows:

Maximize

$$Z = \sum_{j=1}^{n} \sum_{i=1}^{m} c_{ij} \, x_{ij} \tag{5}$$

subject to

$$\sum_{j=1}^{n} a_{ij} \, x_{ij} \le b_i \qquad i = 1, 2, \; . \; . \; . \; , m \tag{6}$$

$$\sum_{i=1}^{m} x_{ij} \le 1 \qquad j = 1, 2, \; . \; . \; . \; , n \tag{7}$$

$$x_{ij} = \begin{cases} 1 & \text{if item } j \text{ is assigned to knapsack } i \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Commonly $c_j = c_{ij}$ for any *ij*, however, (5) – (8) is the more general form. The multi-knapsack problem is not a focus of this research; only MKPs are studied in this dissertation.

1.3 Overview of the Dissertation Research

Many researchers have developed heuristic methods for MKPs. Most competitively test their heuristic against other heuristics or against some common problem set. Often, selected heuristics outperform other heuristics on specific problem sets while they do not perform well on other problem sets. Even though modern heuristic approaches, such as tabu search, genetic algorithms, and simulated annealing (Reeves, 1995) are often employed to obtain a good, albeit sub-optimal, solution in a reasonable amount of time, these algorithms are often applied to MKPs without exploiting knowledge of problem characteristics and are thus not optimized for solution performance.

Few researchers focus on why their approach does well on some problems but not so well on other problems. In short, little has been done to determine why a heuristic performs as it does, despite a call by Hooker (1994) to do exactly this type of research.

The objective of this research is to develop new heuristics and/or improve legacy heuristics for MKPs based on understanding and defining what makes a "best" heuristic. In other words, why do some heuristics work well and other heuristics not work so well? This is difficult to answer since performance varies by type of problem and problem characteristics. This research focuses on constraint tightness, correlation between the objective function and constraints, and inter-constraint correlation. The 2KP is examined first and then the 5KP is studied to answer the question.

Chapter II briefly introduces various solution approaches for MKP and how they solve MKP. The methods presented in Chapter II include a branch-and-bound approach, dynamic programming, greedy heuristics, transformation heuristics, reduction approaches, tabu search, genetic algorithms, simulated annealing, and others. Chapter II presents why the dissertation focuses on greedy and transformation heuristics.

Chapter III provides a background and an empirical analysis regarding legacy greedy heuristics used for 2KP and 5KP. Chapter III shows that the performances of heuristics are affected by various constraint slackness and correlation structures. The competitive test results are examined as a function of problem characteristics. The limitations of the standard benchmark problem sets and how to generate a 5KP set conducive to thorough empirical testing are discussed.

Chapter IV discusses the transformation heuristics based on approaches by Pirkul (1987) and Glover (1977). Greedy heuristics are single pass constructive methods; if variables are set to some value, these values are not changed until the end of the process. Transformation heuristics involve local improvement methods, so each variable varies during the solution process in order to move toward improving solutions. Modern heuristics introduced in Chapter II fall into this category. Chapter IV presents both a literature review and an empirical analysis of the transformation heuristics. The characteristics discussed in this chapter are combined with those of the greedy heuristics to develop the new types of heuristics presented in Chapter V.

Chapter V discusses how to develop or improve heuristics based on knowledge gained via the empirical analysis of the performance of multidimensional knapsack heuristics. Three types of heuristics are introduced: A typed heuristic, new gradient heuristics, and a new reduction heuristic.

Chapter VI shows the improved performance of the newly developed heuristics based on both randomly generated problem sets and benchmark problem sets. Chapter VII summarizes this work's findings, outlines contributions of this work, and identifies areas for further research.

1.4 Contributions of the Dissertation Research

This dissertation research makes four major contributions. The first is that this dissertation shows how the empirical science contributes to development of theory. Hooker (1994) indicated, “Empirical science involves theory”. This research conducts the empirical analysis of heuristics and suggests a theory to develop new heuristics. The second is to develop new types of heuristics optimized for MKP solution performance against all kinds of MKP characteristics. Research to date has yet to provide fundamental rationale for why any one heuristic outperforms another on specific types of MKP instances. The third contribution finds rationale for performance differences and exploits that rationale to develop new heuristic approaches. The fourth contribution provides a new MKP set that includes different characteristics such as constraint slackness, correlation structures, number of variables, and number of constraints. The current standard benchmark MKP set does not provide a sufficiently diverse set of test problems. This means testing using this problem set may not provide enough experimental information regarding the solution performance of heuristics.

The empirical analysis of the legacy greedy heuristics provides insights that lead to new types of heuristics intended to properly respond to constraint slackness settings and correlation structures. More specifically, the results from the empirical analysis present

- the evidence of lack of diversity of the standard MKP test sets;
- method to generate a diverse set of 5KP test problems that vary desired problem characteristics;
- the performance of each heuristic according to various 2KP and 5KP constraint slackness and correlation structures;
- the new knowledge gained via empirical analysis that influences how to develop superior heuristics over all problem characteristics.

The new types of heuristics provide superior and robust results across an entire range of problem instances.

- A typed heuristic applies insights found in the empirical analysis to the generalization of the choice of best heuristic. In other words, choose a legacy heuristic that is mostly likely to produce the best solution among the suite of heuristics (a challenge first proposed by Zanakis (1977) and again by Loulou and Michaelides (1979)).
- New gradient heuristic combines the merits of the better heuristics. All heuristics use different penalty cost functions and handle feasibility differently. The new gradient heuristic includes the favorable influential factors.
- New reduction heuristic combines the new gradient heuristic into the transformation heuristic. The new reduction heuristic is a robust heuristic.

II. Solution Approaches for MKP

2.1 Introduction

The 0 – 1 MKP is an NP-hard problem (Frieze and Clarke, 1984). An NP-hard problem indicates that the worst-case computation time increases rapidly with problem size. Sometimes, exact algorithms such as branch-and-bound require excessively long run times to find an optimal solution. Therefore, heuristic approaches are often employed to obtain high quality sub-optimal solutions, in a reasonable amount of time. However, these algorithms are often applied without exploiting knowledge of problem characteristics and are thus not optimized for solution performance. The details of exact algorithms and heuristics require an introduction into the basics of the corresponding algorithms, and these go beyond the scope of this research. Thus, this chapter briefly presents diverse exact algorithms and heuristics used to solve MKPs.

2.2 Branch-and-Bound Approach

The branch-and-bound (B & B) approach is an exact and possibly exhaustive enumerative algorithm that guarantees finding the optimal solution of a MKP. The B & B approach evaluates (at least implicitly) all feasible solutions and selects the best solution (optimum solution). However, the B & B approach can be impractical in application because the number of feasible solutions could be very large. Some existing heuristics include the B & B approach to improve the heuristic solution.

Thesen (1975) presented an early paper regarding the B & B approach tailored to solve MKP. In his paper, a standard B & B algorithm for MKP was studied with a focus on how to save memory space. Thesen improved the tree structures of the B & B algorithm. Computational tests showed that his algorithm was much faster than the Geoffrion (1967) and Zionts (1972) methods based on test problems with up to 50 variables (*n*) and up to 30 constraints (*m*). However, Thesen's algorithm could not solve a large problem (more than 50 variables) due to the exponential time complexity of the B & B algorithm.

Shih (1979) introduced a modified B & B approach by generating bounds to solve MKP. The general B & B approach requires an objective function value (a bound for objective function value at other nodes) and a choice rule for a branch variable at each node. Shih used greedy heuristic attributes that guarantee optimum solution of the linear programming (LP) relaxation of the KP. He found the optimal solution of the LP-relaxed KP by solving for each of the *m* single constraint relaxed knapsack problems separately, and then used the minimum value among *m* objective function values as a bound for a node. Shih tested his B & B algorithm on randomly generated problems with $m = 5$ constraints and $n = 30$ to 90 variables. He concluded that his algorithm was faster than the standard B & B approach. His B & B algorithm averaged 13.385 minutes while the standard B & B approach averaged more than 380.755 minutes.

Balas and Zemel's algorithm (1980) embedded the B & B approach based on the optimal solution of the LP relaxation to solve KP. Some variables have fractional solutions in the optimal solution of the LP relaxation. They approximated a subproblem which included these variables, and then applied the B & B approach to find the optimal solution of this subproblem.

Gavish and Pirkul (1985) combined a transformation approach with the B & B approach. More recently, Martello and Toth (2003) combined a greedy heuristic and a transformation approach with the B & B approach. Both algorithms reduced the problem size with a transformation approach and a greedy heuristic, and then applied B & B to the small problems.

Even though the B & B approach has limited application to the MKP due to the problem's exponential time complexity, the most important merit of the B & B approach is its guarantee to find the optimal solution.

2.3 Dynamic Programming

Two standard approaches, dynamic programming and the branch-and-bound algorithm, guarantee optimum solutions of MKPs. Dynamic programming can find the optimum solutions of MKP by solving small subproblems of the MKP and then expanding this solution approach by solving another subproblem based on the previous solution in an iterative fashion until the complete MKP is solved. Dynamic programming is also an exhaustive enumeration approach as it may consider all 0 – 1 combinations of n variables. Dynamic programming's main concept can be stated as follows:

$$f(k, \mathbf{b}') = \max_{x_k \in \{0,1\}} \{c_k x_k + f(k-1,\ \mathbf{b}' - \mathbf{A}_k)\} \quad \text{for } k = 1, \ldots, n \tag{9}$$

subject to $\qquad \mathbf{b}' - \mathbf{A}_k \geq \mathbf{0} \quad \text{and} \quad f(0, \mathbf{b}') = 0$

where $f(k, \mathbf{b}')$ is the total objective function value of the first k variables in solution, $\mathbf{b}'$ is the remaining RHS vector by inclusion of first k variables, and $\mathbf{A}_k$ is the constraint coefficient vector of the kth variable. Equation (9) is repeated until all items are considered, *i.e.,* $k = 1, \ldots, n$. The value of $f(n, \mathbf{b}')$ is the optimal solution of MKP.

Weingartner and Ness (1967) introduced a dual approach for dynamic programming. In a dual approach, dynamic programming starts with all items included in the knapsack, *i.e.*, infeasible solution, while a primal dynamic programming approach starts with an empty knapsack. In the dual approach, the dynamic programming approach proceeds to remove items until feasibility is achieved. They also presented a methodology for reducing problem size through use of lower and upper bounds that were determined by heuristics.

Soyster *et al.* (1978) introduced an enumerative technique based on the optimal solution of the LP-relaxed MKP. They found the optimum solution of the LP-relaxed MKP and then partitioned this solution into two sets, one including all fractional solutions and the other including all integer solutions. They found the optimum integer solution of the subproblem using an enumerative technique in which the subproblem consisted of only fractional valued variables. At each iteration, the subproblem's size was increased by adding one variable. The termination criteria is that the current solution differs by a value of less than 1 from the previous best solution. Thus, their algorithm is an approximate algorithm that dynamically decreases the number of solutions compared.

Bertsimas and Demir (2002) presented an approximate dynamic programming approach for the MKP. They initiate dynamic programming with a subproblem composed of the first k variables, rather than just the first variable. They used several greedy heuristics to approximate the optimal value of the initial subproblem, $f(k, \mathbf{b}')$ in Equation (9), and then extended this heuristic solution to construct the next suboptimal solution $f(k+1, \mathbf{b}')$ by a standard dynamic programming approach. To choose the best greedy heuristic to approximate the suboptimal solution, they based a comparison of six different greedy heuristics, including their own adaptive fixing heuristic, on computational tests. Their own heuristic is fast and accurate. They recommend that these greedy type heuristics are worthy of future investigation.

In addition to the above references, Weingartener and Ness (1967), Psinger (1997) and Martello *et al.* (1999) employed dynamic programming to solve the KP. They reduce computational time and memory space by incorporating other types of algorithms.

This section presented exact approaches, the branch-and-bound approach and dynamic programming, to find the optimum solutions of MKPs. Since there are huge computational difficulties associated with both approaches, researchers have tried to incorporate these exact algorithms with other heuristics to reduce the number of comparisons or the size of problems in order to improve computation time and storage space. However, the inclusion of heuristics into the exact algorithms involves a tradeoff between the optimum solution and the approximate solution of MKP.

2.4 Greedy Heuristics

Many effective greedy solution procedures for the MKP have been developed; Senju and Toyoda (1968), Kochenberger *et al*. (1974), Toyoda (1975), Loulou and Michaelides (1979), Fox and Scudder (1985), and Lee and Guignard (1988) are examples. A greedy heuristic repeatedly takes the best immediate or local step or move, to find a solution. Thus, a greedy heuristic arrives at a solution by making a sequence of choices, each of which is made by selecting the best choice at the particular moment. Greedy type heuristics can be distinguished as using either a primal (Toyoda (1975), Loulou and Michaelides (1979), Kochenberger *et al.* (1974), Fox and Scudder (1985), Lee and Guignard (1988)) or a dual approach (Senju and Toyoda (1968)). A primal approach starts with an empty knapsack (all variables are set equal to zero), and items are added (set variables equal to one) according to a given rule while not violating feasibility. A dual approach starts with all items in the knapsack, (all variables are set equal to one), and then removes items from the knapsack (sets variables equal to zero) according to a given rule until feasibility is achieved.

The details of greedy approaches and the solution performances of greedy approaches by Senju and Toyoda (1968), Kochenberger *et al*. (1974), Toyoda (1975), Loulou and Michaelides (1979), and Fox and Scudder (1985) are examined in Chapter III.

Lee and Guignard (1988) presented a greedy heuristic that consisted of two phases: for Phase I, their heuristic is incorporated with Toyoda's heuristic (1975) to find a feasible solution and fix variables for reduction of problem sizes, and for Phase II, they improve the current solution using a complementing procedure from Balas and Martin (1980). Computational tests indicated that their heuristic produced better solution quality than that of Toyoda (1975) and Magazine and Oguz (1984), but the solution quality was worse than the solution quality of Balas and Martin (1980).

Greedy heuristics often are used as a base heuristic to improve the solution performance of exact algorithms and other heuristics. Papers related to the greedy type heuristics are numerous.

2.5 Transformation Heuristics

KP is a special case of the MKP (m =1). Exact algorithms and heuristics for KP are more studied and well developed than for MKP. A transformation heuristic changes the MKP to a KP using approaches such as Lagrangian relaxation, surrogate relaxation, or composite relaxation. An example and explanation to illustrate transforming MKP to KP based on Lagrangian relaxation and surrogate relaxation and the detailed solution performance of transformation heuristics by Glover (1977) and Pirkul (1987) are provided in Chapter IV. Composite relaxation is a combination of Lagrangian relaxation and surrogate relaxation.

Magazine and Oguz (1984) introduced a transformation heuristic, Multi-Knap, by combining the dual approach of Senju – Toyoda (1968) with Everett's generalized Lagrange multiplier approach (1963). Multi-Knap transforms MKP to KP using Lagrange multipliers, and then selects a variable based on a profit/cost ratio of the transformed KP. In their computational tests, Multi-Knap was compared to the heuristics of Senju and Toyoda (1968) and Kochenberger *et al*. (1974). The Kochenberger *et al*. heuristic was found better than Senju – Toyoda and Multi-Knap in terms of solution quality. However, from a computation time perspective, Multi-Knap was better than the other heuristics examined. Volgenant and Zoon (1990) analyzed and improved Magazine and Oguz's heuristic (1984) by modifying how to compute the Lagrange multipliers. This modification yielded an improvement in the computing time and solution quality of Magazine and Oguz's heuristic.

Gavish and Pirkul (1985) tested various relaxations of the MKP including Lagrangian, surrogate and composite relaxation in order to find the lowest upper bound for the MKP. Their computational experiments and comparisons indicate the composite relaxation provides the best overall bounds. They suggest rules for reducing the problem sizes based on surrogate relaxation, then solve the reduced problem using the modified branch-and-bound algorithm. Their algorithm was compared to the branch-and-bound method of Shih (1979) in terms of computing time, and compared to the greedy heuristic of Loulou and Michaleides (1979) in terms of solution quality. Their algorithm yielded faster computing time and better solution quality.

2.6 Reduction Approaches

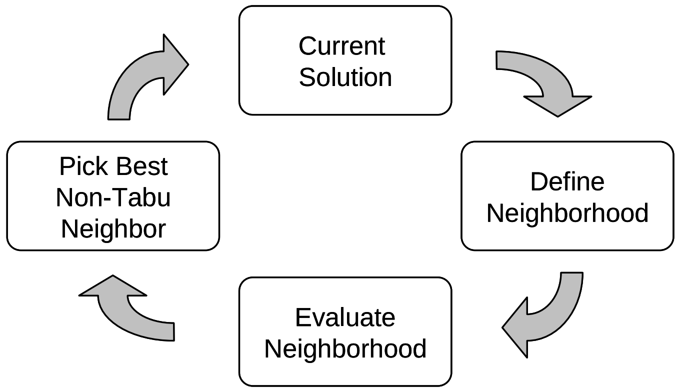
If the modified MKP has fewer variables and constraints than the original MKP, and the modified MKP can have the same optimal solution as the original MKP, it is clearly advantageous to reduce the size of the MKP. The modification of MKP can occur two ways: (1) reducing the number of variables (fix the values of some variables before the actual solution procedure is started) and (2) reducing the number of constraints (transform MKP to KP (multi-constraints to single constraint)). Each reduction has the benefit of a smaller problem instance for all solution procedures such as exact algorithms and heuristics.

Fréville and Plateau (1986, 1994) introduced a reduction approach, RAMBO, in terms of variable and constraint eliminations. RAMBO fixes the value of selected variables and eliminates constraints by the conjunction of simple tests applied to 0 – 1 knapsack problems derived from the original problem. Their reduction procedure decreases the size of the problem, and improves the efficiency of enumerative methods. Their algorithm also provides upper bounds using Lagrangian relaxation and surrogate relaxation and lower bounds using two greedy heuristics.

Osorio *et al*. (2002) presented constraint pairing as a reduction algorithm for MKP. Considering all constraints of MKP usually yielded better solution quality than considering a single constraint one at a time. However, the computational effort to consider all constraints of MKP can be excessive. A constraint pairing is "a compromise between the two extremes: examining the full constraint system on one hand, and examining individual constraints on the other" (Osorio *et al*., 2002). Using a constraint pairing, they fix some variables to zero and separate the rest into two groups – those that tend to be zero and those that tend to be one in the optimal solution. These reduction procedures increase efficiency of the branch-and-bound algorithms with fewer nodes than commercial software.

2.7 Tabu Search

Tabu search (TS) is one of the most popular modern meta-heuristics. The concept of the tabu search was developed in the 1970s; however in 1986, Glover first presented the current form of tabu search (Glover, 1977, 1986). There are two important elements in tabu search: one is the moves and the other is the tabu list. The "moves" define a neighborhood which is the set of candidate solution states reachable from the current solution state in one set of solution characteristic changes. The tabu list stores the last few solution states visited and restricts their being revisited. The basic concept of the tabu search, as described by Glover (1986), is to avoid cycles by forbidding moves that take the solution to points in the solution space previously visited. Figure 1 shows a general iteration in a tabu search.



(Harder *et al.*, 2004)

Figure 1. An Iteration in a Tabu Search

Glover (1986) states, "tabu search is a meta-heuristic superimposed on another heuristic". Thus, a tabu search requires a subroutine, another heuristic, to find a local optimum. When this subroutine finds a local optimum, the tabu search moves from the current solution space to another solution space, to find another local optimum. Many different tabu searches have been created by using different heuristics to find local optima.

Aboudi and Jörnsten (1994) applied a tabu search to MKP using the pivot and complement heuristic of Balas and Martin (1980) as a subroutine. Aboudi and Jörnsten defined "moves" as the better integral solution found by the pivot and complement heuristic, and defined "tabu list" as the last integral solution found by the pivot and complement heuristic. This tabu condition is enforced by adding a constraint to the last formulation preventing the pivot and complement heuristic from retuning to previously found solutions. To explore different solution states, they used different initial inputs for the pivot and complement heuristic to escape from local optimal solutions.

Glover and Kochenberger (1996) introduced tabu search combined with strategic oscillation. The concept of strategic oscillation was first presented in Glover (1977). Their strategic oscillation process alternated between a constructive phase and destructive phase based on the current solution's feasibility. They defined a "critical event" as the last feasible solution found after a transition between phases. Their tabu search focused on those critical events which drove the search to variable depths on each side of the feasibility boundary in order to improve solution quality.

Løkketangen and Glover (1998) used tabu search to solve general mixed integer programming (MIP) problems, while Aboudi and Jörnsten (1994) only focused on the 0 – 1 MKP. However, Løkketangen and Glover also conducted an empirical test of their tabu search on 0 – 1 MKPs. Løkketangen and Glover used a standard bounded variable simplex method as a subroutine to exploit the fact that an optimal solution to the 0 – 1 MIP problem may be found at an extreme point of the LP feasible set. The tabu search by Løkketangen and Glover is as follows:

Step 0: Begin by solving the LP relaxation of the zero-one MIP problem to obtain an optimal LP solution. x^* is the best MIP feasible solution found and z^* is its objective function value.

Step 1: From a current LP feasible basic solution, consider the feasible pivot moves that lead to adjacent basic feasible solutions.
(a) Isolate and examine a candidate subset of these feasible pivot moves.
(b) If a candidate move creates an MIP feasible solution x whose associated objective function value z yields $z > z^*$, record x as the new x^* and update z^*

Step 2: Select a pivot move that has the highest evaluation from those in the candidate set, applying TS rules to exclude or penalize moves based on their tabu status.

Step 3: Execute the selected pivot updating the associated TS memory and guidance structures, and Return to Step 1.

(Løkketangen and Glover, 1998)

To define “moves”, Løkketangen and Glover introduced four types of move strategy: (1) Type I, decreasing profit and decreasing infeasibility; (2) Type II, increasing profit and increasing infeasibility; (3) Type III, non-decreasing profit and non-increasing infeasibility; (4) Type IV, decreasing profit and non-decreasing infeasibility. To define “tabu status,” they tracked two pieces of information, the current solution status and solution frequency. Even though their approach was developed for MIPs, their computational testing was employed on MKP test sets. They achieved results comparable to solutions obtained by their own simple greedy heuristic, which was developed to take advantage of the special structure of the MKP.

Hanafi and Fréville (1998) introduced a different tabu search approach for 0 – 1 MKPs by using the concept of greedy heuristics within a tabu search framework. Hanafi and Fréville introduced an oscillation strategy which emphasizes a search on the boundary between feasibility and infeasibility because the optimal solution is usually located on the boundary or close to the boundary. Their “moves” are exchange 0 and 1 values of variables regardless of feasibility, and their “tabu list” stores the current best solution and all the transitions performed. To drive the search into new regions of the search space, Hanafi and Fréville used simple greedy heuristics to select variables to drop or add. If the current region is a feasible region, they add the variable with the maximum bang-for-buck ratio (profit/cost) to move to an infeasible region. They showed that tabu search based on oscillation strategy can efficiently solve MKPs by effectively searching on the boundary between feasibility and infeasibility.

The most recent references to tabu search for MKPs are Vasquez and Hao (2001) and Oppen *et al.* (2003). Vasquez and Hao incorporate linear programming into a tabu search. Their suggestion is that the search space around the optimum of the LP relaxation problem of the MKP should contain high quality solutions. Thus, they solved the LP-relaxed MKP, used a fractional optimal solution as an initial starting solution for the tabu search, and then used a tabu search to explore areas around these fractional optimum points. The tabu search by Oppen *et al.* (2003) is very similar to the Hanafi and Fréville (1998) heuristic as it uses a greedy heuristic within a tabu search framework.

Tabu search for MKPs is summarized as follows: (1) it usually requires a heuristic to find a good feasible starting solution, (2) it usually defines "moves" as the flip of variables which means assigning the opposite value to a variable (*i.e.*, assign zero to a variable which has value of one), (3) it requires a general heuristic or exact algorithm as a subroutine to find a local optimum or evaluate a current solution, (4) it requires recency and frequency information for "the tabu list" where recency information stores a recently flipped variable to avoid cycling and frequency information stores the number of flips of a variable to drive the search into different solution spaces where solutions have not been visited.

2.8 Genetic Algorithm

The genetic algorithm (GA) is a class of meta-heuristics which imitate the natural genetic process of biological organisms. In 1975, John Holland first introduced the genetic algorithm (Holland, 1975). The genetic algorithm begins with a group of solutions called a population. Each solution in the population is evaluated by the fitness function to select a subset of the solutions to be either used directly in the next population (reproduction) or transformed (mutation or crossover). The mutation assigns an opposite value to a randomly chosen variable ($0 \rightarrow 1$, or $1 \rightarrow 0$). The crossover chooses two solutions, and then combines these solutions to produce new solutions. Unlike other meta-heuristics, the genetic algorithm includes probability to select solutions for the next generation. Better solutions in the current population will have a higher probability to reproduce a solution for the next generation. The best solution to date is usually recorded. Figure 2 shows an overview of the genetic algorithm.

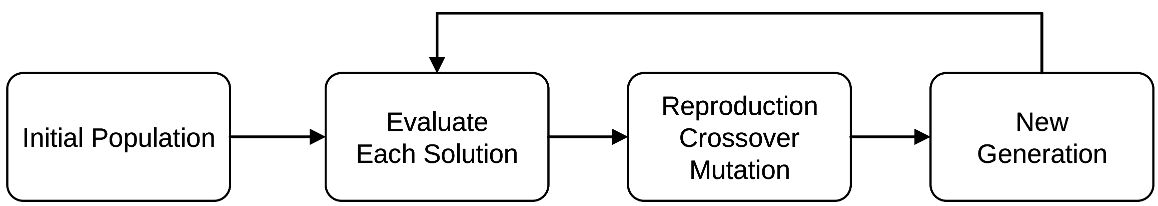


Figure 2. Overview of Genetic Algorithm

Khuri *et al.* (1994) introduced a genetic algorithm to solve the MKP using the software package GENEsYs. Since the GENEsYs package has flexibility regarding a fitness function and genetic operator (such as the function for probability of reproduction, crossover, and mutation), Khuri *et al.* found the proper parameter setting for GENEsYs. Khuri *et al.* initiated the genetic algorithm with a uniform randomly generated initial population (size 50), a mutation probability of 1/ 50, and crossover probability of 0.6. For crossover, the genetic algorithm chose two solutions at random from the current population, chose a crossover point between *i*th and *j*th variables at random, and exchanged all bits after the *i*th variables between both solutions. Khuri *et al.* used a simple fitness function with a graded penalty term for reproduction. Unlike most uses of genetic algorithms, infeasible solutions are allowed to participate in the search, so the genetic algorithm has variation in creating the next generation. For termination criteria, a total of 100 runs are used. Khuri *et al.* suggested that proper parameter selection and a simple fitness function accommodating infeasibility made the genetic algorithm more effective in solving MKPs.

Theil and Voss (1994) compared three types of genetic algorithms: (1) a standard genetic algorithm, (2) a modified genetic algorithm integrating simple heuristic operators with an integer coding, and (3) a hybrid-GA based on combining a genetic algorithm with a tabu search. The modified genetic algorithm starts with a randomly generated population. For infeasible solutions in the population, an ADD/DROP heuristic is employed to remove variables until feasibility is achieved. The objective function is used as the fitness function. If infeasible solutions are allowed, the objective function value of an infeasible solution is reduced by some penalty function. Mutations include the DROP-ADD operator which starts by dropping variables, and in the next step, other variables are added while maintaining feasibility. For a hybrid-GA, the tabu operator is employed to avoid local optimality. Theil and Voss compared three algorithms on 57 literature test problems and concluded: (1) the standard GA cannot obtain good solutions, (2) a modified GA got acceptable results when applied to small problems, results that were much better than those achieved by the standard GA, and (3) the results obtained by a hybrid-GA were better than results by other meta-heuristics, such as simulated annealing or tabu search.

Hoff *et al.* (1996) discussed how to find good parameter settings and search mechanisms for the genetic algorithm based on empirical analysis. Hoff *et al.* conducted empirical testing to optimize parameters such as population size, mutation operator, crossover function, population type, fitness function, and new generation size. For population size, they found that too small a population provides too little diversity, while too large a population causes the genetic algorithm to require long computing times. Their empirical tests found that a population size of 5*n* worked best, where *n* is the number of variables. Among different mutation operators, an inverted mutation, which gave the opposite value to a selected variable, performed best with a mutation rate of $\frac{1}{n}$. For a crossover function, the burst crossover was the best, in which a crossover is made at every bit position in turn with 0.5 probabilities, thus possibly resulting in many crossover points. For population types regarding either feasibility or infeasibility, the population that always maintained feasibility worked best. Hoff *et al.* examined three different fitness functions: objective function value, the fitness function by Khuri *et al.*, and the objective function value minus the square root of the sum of overflow of the knapsack. They found that none of the above fitness functions was adapted to a feasible population type. The fitness function of Khuri *et al.* is population type – specific, thus it did not perform well with a feasible population type. The number of generations should be 50 times the population. Hoff *et al.*, in their conclusion, noted how careful parameter tuning was very successful, and thus the genetic algorithm can yield superior results.

Chu and Beasley (1998) combined a genetic algorithm with the greedy heuristic portion of Pirkul's heuristic (1987). Chu and Beasley's genetic algorithm only considers feasible solutions. Thus, if an infeasible solution is created, they drop variables according to a simple effective gradient value (G_j) of the greedy heuristic until feasibility is achieved. This operation is called "a repair operator" and guarantees all solutions are feasible. Their default setting for the genetic algorithm includes the binary tournament selection method (two solutions randomly selected for parents from two pools), the uniform crossover operator (each bit in the new generation is created by copying the corresponding bit from one or the other parent solution), and a mutation rate equal to 2 bits per child string. Chu and Beasley compared their genetic algorithm performance to heuristics by Magazine and Oguz (1984), Volgenant and Zoon (1990) and Pirkul (1987). Their genetic algorithm outperformed the other heuristics in terms of solution quality.

Raidl (1998) presented an improved genetic algorithm for the MKP by pre-optimization of the initial population. Raidl suggested that the solution quality of a genetic algorithm could be improved with an initial population of pre-optimized solutions determined by a greedy heuristic based on the LP-relaxed MKP and using probabilistic repair operators and local improvements to create the next generation. Since the solution values of the LP-relaxed MKP are fractional values from 0 to 1, Raidl used these fractional values as probabilities for setting variables equal to one in the initial population. The "repair operator" is similar to Chu and Beasley's (1998); the only difference is that Raidl's method is probabilistic based on the solution of the LP-relaxed MKP, while Chu and Beasley's method is deterministic. Computational tests indicated that Raidl's genetic algorithm outperformed Chu and Beasley's genetic algorithm on solution quality and faster computing time based on Beasley's standard MKP test set.

Two points can summarize genetic algorithms for the MKP: First, proper parameter settings for the genetic algorithm are very important in obtaining better MKP solution quality. Second, to improve the final solution quality of the genetic algorithm, apply another heuristic to select better solutions in the current population or incorporate it in the optimization steps of the genetic algorithm to create a better next generation.

2.9 Simulated Annealing

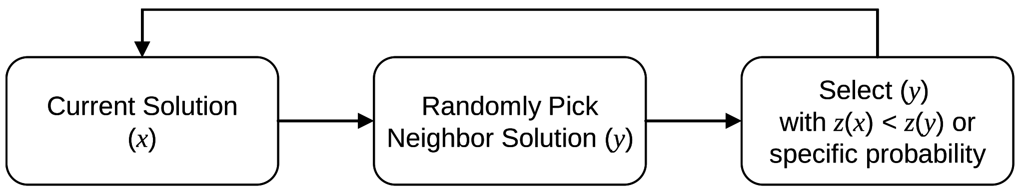
Simulated annealing is another popular modern meta-heuristic. As with other meta-heuristics, simulated annealing is a local search technique designed to allow it to escape from local optima. Simulated annealing imitates a physical annealing process applied to solids, where a material is heated into a liquid state then cooled to a recrystallized solid state. Kirkpatrick *et al.* (1983) showed that the physical annealing simulation is applicable to optimization problems. Figure 3 shows the mapping of physical annealing terminology to simulated annealing terminology for an optimization problem.

Thermodynamic Simulation	Combinatorial Optimization
System State	Feasible Solution
Energy	Cost
Change of State	Neighboring Solutions
Temperature	Control Parameter
Frozen State	Heuristic Solution

(Kirkpatrick *et al.* (1983))

Figure 3. Physical Annealing vs. Simulated Annealing Terminology

Like other meta-heuristics, simulated annealing starts with an initial feasible solution, evaluates candidate (neighborhood) solutions, and selects one of the candidate solutions that improves the objective function value or escapes from local optima. There is big difference between simulated annealing and other meta-heuristics as simulated annealing uses more probabilistic parameters in its approach. This probabilistic attribute allows escape from a local optimum while tabu search uses a deterministic approach to escape a local optimum. Figure 4 shows an overview of an iteration in simulated annealing.



z: objective function value

Figure 4. An Iteration in Simulated Annealing

Drexl (1988) applied a simulated annealing approach to MKP, an approach called PROEXC (probabilistic exchange). The basic procedure of PROEXC is (1) to start with a random feasible solution, (2) to categorize two sets of variables: one set includes all variables equal to zero and the other set includes variables equal to one, (3) to randomly choose one variable from each set, (4) to exchange two variables if objective function value increases or subject to a given probability, and (5) conduct a given number of iterations at each temperature. Drexl used 57 test problems to determine the best parameters of the simulated annealing. Computational tests indicate PROEXC provides equally fast and equally good solutions compared to the heuristic of Gavish and Pirkul (1985).

2.10 Others

Hillier (1969) introduced a heuristic for MKP based on the simplex method. His heuristic consists of three phases: Phase 1 identifies initial solutions which are the optimal noninteger solution obtained by the simplex method and nearby integer solutions, Phase 2 searches nearby integer solutions, and Phase 3 attempts to improve the feasible solution obtained in Phase 2.

Balas and Martin (1980) suggested the "pivot and complement" heuristic for 0 – 1 KP. It uses the fact that a 0 – 1 KP is equivalent to the associated LP with the added requirement that all slack variables are basic. Their heuristic starts by solving the LP, performs a sequence of pivots aimed at putting all slack variables into the basis at a minimal cost, and finally improves solution quality using a local search based on complementing certain sets of 0 – 1 variables.

Glover (1994) proposed a heuristic based on a neural network concept, "ghost image processes". The results obtained by ghost image processes were compared to those obtained by Senju and Toyoda's heuristic (1968). Glover's heuristic yielded better solution quality on randomly generated problems than Senju – Toyoda's, albeit its computational time was slower than Senju – Toyoda's algorithm.

Averbakh (1994) examined dual properties of the MKP for different probabilistic models. Averbakh presented a heuristic based on Lagrangian relaxation and these probabilistic models. Since Lagrangian relaxation provides upper bounds to the MKP, the probabilistic models provide good Lagrangian multipliers, which are the control parameters of the heuristic, to reduce the gap between Lagrangian relaxation solution and the optimal solution of MKP.

2.11 Summary

The MKP is an important and difficult class of combinatorial optimization problem. Various heuristics and exact methods have been developed to solve MKP. While, there are many variants within a given method, these methods are general and each method has its own preferred parameter values for specific problem characteristics. Although the efficiency of an algorithm has been demonstrated in computational tests, their use may be limited due to their specificity. Thus, in practice, when using such methods to solve MKP, one encounters the dilemma of which heuristic or exact method to select, and then, within the selected method, which of the variants to implement.

As shown above, it is a common and popular scheme to combine different heuristics: Shih (1979), Balas and Zemel (1980), and Martello and Toth (2003) combined a greedy heuristic with a branch-and-bound algorithm; Weingartner and Ness (1967), and Bertsimas and Demir (2002) dynamic programming; Magazine and Oguz (1984), and Volgenant and Zoon (1990) transformation approaches; Aboudi and Jörnsten (1994), Glover and Kochenberger (1996), Hanafi and Fréville (1998), and Oppen *et al.* (2003) tabu search; Chu and Beasely (1998) and Raidl (1998) genetic algorithms.

Greedy heuristics provide good bounds for exact methods, find good initial feasible solutions, and improve solution qualities of modern heuristics. Many researchers have insisted that they have improved the main method for MKP simply by using different greedy heuristics as a base heuristic. Even though they have showed the competencies of their methods based on computational tests, the test problems are overly specific. Thus, their method might fail to produce expected solution quality when dealing with other types of problem characteristics because heuristics for finding solutions tend to be problem specific. Therefore, in this research, greedy heuristics and transformation approaches are examined to better understand how they perform as a function of problem characteristics, thereby yielding more robust greedy and transformation heuristics. Greedy heuristics are examined in the next chapter.

III. Empirical Analyses of Legacy Greedy Heuristics

3.1 Introduction

This chapter studies the legacy greedy heuristic methods proposed by Toyoda (TOYODA, 1975), Senju – Toyoda (S – T, 1968), Loulou – Michaelides (L – M M1, M2, SW1, SW2, 1979), Fox and Scudder (FOX, 1985), and Kochenberger *et al.* (KOCHEN, 1974). The background section provides an overview of empirical analyses and the heuristic solution procedures for multidimensional knapsack problems. The next section looks at the lack of diversity of the standard problem set, describes the characteristics of the benchmark problem set, and discusses its weaknesses. Finally, empirical analyses of the heuristic solutions of 2KP and 5KP test problem sets illuminate the effects of constraint slackness and correlation structure on the performance of heuristics.

3.2 Background

3.2.1 Empirical Analyses of Heuristics

Hooker (1994) suggested that the performance of algorithms be analyzed in two ways: one is to analyze performance analytically relying on the methods of deductive mathematics and the other is to analyze performance empirically using computational experiments. In the operations research literature, deductive mathematical methods are more developed than empirical analysis. However, the mathematical approach does not usually indicate how an algorithm is going to perform on typical problems. If we want to know how an algorithm works on typical problems, computational experiments give much more insight into algorithmic performance.

Through empirical analysis, it can be determined how algorithms work and why algorithms perform well or poorly. Thus, algorithms should be analyzed to gain insight into theory as insights from empirical analysis might suggest theory. Since a heuristic method is an algorithm, a heuristic can be analyzed through empirical testing (Hooker, 1994).

Computational complexity theory classifies problems according to their solution difficulty based on a measure of worst-case running time. There are two basic divisions of problems: those problem classes easy to solve as there exists a provable polynomial time algorithm for the problem class, and those problems difficult to solve as there is no provable polynomial time algorithm for the problem class. The latter are referred to as the problems in the Class NP (Parker and Rardin, 1982).

NP Problems are very difficult to solve and sometimes, impossible to solve in a reasonable amount of time. Heuristics, however, offer a way to find reasonable solutions in a reasonable amount of time. The solutions typically come with a feasibility guarantee but are not guaranteed optimal. The quality of a solution can be used to compare heuristics. The next section introduces greedy MKP heuristics.

3.2.2 Legacy Greedy Heuristics for MKP

The MKP is encountered when one has to decide how to choose items to satisfy multiple resource constraints. As shown in Chapter II, many effective greedy solution procedures for the MKP are used independently or incorporated into exact algorithms or meta-heuristics to improve the performance of those approaches. This section examines the greedy heuristics proposed by Toyoda (1975), Senju and Toyoda (1968), Loulou and

Michaelides (1979), Kochenberger *et al*. (1974), and Fox and Scudder (1985). Each greedy heuristic uses a penalty cost (v_j) to quantify the relative worth of the items.

Toyoda's Heuristic (TOYODA)

Toyoda's (1975) approach is to evaluate each item in terms of its effective gradient, a measure of item worth per unit cost in terms of all resources used. Toyoda first normalized each constraint, so all right-hand side (RHS) values were 1 by multiplying the ith constraint in Equation (3) by $\frac{1}{b_i}$. The Toyoda penalty cost function is defined as follows:

$$v_j = \sum_{i=1}^{m} \frac{a_{ij}\, b_i^0}{\sqrt{\sum_{i=1}^{m} (b_i^0)^2}}, \; j = 1, \ldots, n \tag{10}$$

where b_i^0 is the resource used in constraint i, and $0 \le b_i^0 \le 1$ for $i = 1,\ldots, m$ where m is the number of constraints. Then, the effective gradient is computed for each item as

$$G_j = \frac{c_j}{v_j}, \quad j = 1, \ldots, n \tag{11}$$

where c_j is the objective function coefficient for the jth item and the value v_j is updated each iteration. Toyoda's method starts with an empty knapsack and then adds the item with the highest scoring feasible effective gradient until some knapsack constraint prevents further placement of items in the knapsack.

Toyoda (1975) called this a primal effective gradient method because the solution is always feasible. An overview of this approach is as follows:

Step 1: Start with all items designated as not contained in the knapsack.

Step 2: Compute the effective gradient for each candidate item not currently in the knapsack and feasible.

Step 3: Order the candidate items in descending order of their effective gradient measures.

Step 4: Add highest scoring item to the knapsack.

Step 5: Go to *Step* 2, maintaining problem feasibility across all constraints.

Senju and Toyoda's Heuristic (S – T)

Senju and Toyoda's (1968) method starts with all items designated as contained in the knapsack. The heuristic then drops items (x_j) according to the ascending order of a dual effective gradient until feasibility is achieved. Their approach is a two-pass algorithm since the authors realized the computed gradients might cause the algorithm to "over shoot" the feasibility target. Thus, once feasible, the Senju – Toyoda method attempts to restore items previously dropped. To define their penalty cost (v_j), let $P_j = \{a_{1j}, a_{2j}, \ldots, a_{mj}\}$ be the vector of constraint coefficients for item j of n items. Let $T = \{\sum_{j=1}^{n} a_{ij}, i = 1, \ldots, m\}$ be the vector of total resources consumed in each of m constraints when all items are in the knapsack, and $L = \{b_1, b_2, \ldots, b_m\}$ be the vector of right – hand side values, so $R = T – L$ is the surplus vector. The Senju – Toyoda penalty cost function is defined as follows:

$$v_j = \frac{P_j \cdot R}{|R|}, \; j = 1, \ldots, n \quad (12)$$

where $|R|$, the length of *R*, is defined as $|R| = \sqrt{r_1^2 + r_2^2 + \cdots + r_m^2}$. Consequently, a dual effective gradient is: $G_j = \frac{c_j}{v_j} = c_j\left(\frac{|R|}{P_j \cdot R}\right)$.

An overview of this dual approach follows:

Step 1: Start with all items designated as contained in the knapsack.

Step 2: Compute a dual effective gradient for each item.

Step 3: Order the items in ascending order according to their dual effective gradient measures.

Step 4: Remove items with the lowest dual effective gradient measure until feasibility with respect to all constraints is achieved.

Step 5: Re-consider removed items for inclusion if all constraints are non-binding.

Loulou and Michaelides' Heuristic (L – M)

Loulou and Michaelides' (1979) expand on Toyoda's (1975) approach. If two candidate items have an equal v_j, select the item which consumes the least actual amount of the resources. They proposed four ways to calculate the penalty cost v_j. To calculate their effective gradient, three important concepts are introduced.

$DA_i + a_{ij}$: total consumption of resource *i* if x_j is added to the current solution (set to one) where DA_i is the amount of resource *i* consumed so far.

$1 - (DA_i + a_{ij})$: amount of resource *i* remaining if x_j is added to current solution (set to one)

$\sum_{k \in SC} a_{ik} - a_{ij}$: future potential demand for resource *i* if x_j is added to current solution (set to one) where *SC* is the set of candidate variables

$(x_j = 0,\ j \in SC)$.

The first method for calculating penalty factor v_j is then:

$$v_j = Max_{i=1,\dots,m}\{(DA_i + a_{ij})\left(\sum_{k \in SC} a_{ik} - a_{ij}\right)/(1 - DA_i - a_{ij})\}. \quad (13)$$

The second method decreases the importance of the ratio in calculating v_j by using the square root function:

$$v_j = Max_{i=1,\dots,m}\{(DA_i + a_{ij})\left(\sum_{k \in SC} a_{ik} - a_{ij}\right)^{1/2}/(1 - DA_i - a_{ij})^{1/2}\}. \quad (14)$$

The third and fourth methods are based on the first and second methods, respectively. These approaches use v_j until Max$_i$ DA_i becomes "close enough" to 1 and, from then on, selects items according to c_j values only. This modification is called a switch. The switch is actuated when some resource becomes so scarce as to suggest that the algorithm is close to terminating.

Kochenberger *et al*. Heuristic (KOCHEN)

Kochenberger *et al.* (1974) developed a primal heuristic based on the Senju – Toyoda heuristic. Kochenberger *et al.* applied a surplus vector indicating how much resource remained, a concept originally introduced by Senju and Toyoda (1968).

The outline of the KOCHEN heuristic procedure is as follows:

Step 1: Set $x_j = 0, j = 1, \dots, n$

Step 2: Compute $\overline{b_i} = b_i - \sum_{j=1}^{n} a_{ij}x_j$ for all i; where $\overline{b_i}$ is remaining resource in the ith constraint

Step 3: Compute $\bar{a}_{ij} = \dfrac{a_{ij}}{\bar{b}_i}$; where $\bar{a}_{ij}$ for all i, $j \in SC$ is the portion of remaining resource i consumed if variable j is selected

Step 4: Compute $T_j = \sum_{i=1}^{m} \bar{a}_{ij}$ for $j \in SC$.

Step 5: Compute $\bar{c}_j = c_j / T_j$ for $j \in SC$; $\bar{c}_j$ is the primal effective gradient.

Step 6: For $\max_{j \in SC} \bar{c}_j$, set $x_j = 1$, remove j from SC.

Step 7: Go to *Step* 2, maintaining problem feasibility across all constraints.

Fox and Scudder's Heuristic (FOX)

Fox and Scudder (1985) presented a basic structure of primal and dual greedy heuristics for MKPs, and proposed a penalty cost heuristic based on a primal approach applied to a generalized set-packing model. A generalized set-packing model is a specific form of a MKP with the additional requirement that $c_j = 1$ and $a_{ij} = 0$ or 1, for all i and j. Thus, the FOX heuristic can be applied to a pure 0 – 1 MKP with $c_j > 0$, and all $a_{ij} \geq 0$ (at least one $a_{ij} > 0$ for each j). They suggest that general effective gradient rules require that quantities G_j be computed for each $j \in V$ as

$$G_j = \frac{c_j}{\sum_{i \in C} w_i a_{ij}} \tag{15}$$

where V is the set of all variables, C is the set of all constraints, and w_i represents a "weight" assigned to constraint i reflecting its " importance" based on the relative "scarcity" of resource i. In a primal heuristic, variable x_k is set to 1 when G_k equals the maximum of G_j over $j \in V$. In a dual heuristic, variable x_k is set to 0 when G_k equals the minimum of G_j over $j \in V$. The weights are updated in successive iterations to reflect changing constraint importance or resource scarcity as variables are set to 1 or 0.

In the FOX heuristic, the weight is defined as follows:

$$w_i^p = \begin{cases} 1 & \text{if } b_i^p = b_{\min}^p \\ 0 & \text{otherwise} \end{cases} \qquad \text{for all } i \tag{16}$$

where b_i^p is the amount of resource i remaining at the end of iteration p and $b_{\min}^p = \min\{b_i^p\}$ for all i. Thus, in the FOX heuristic, a constraint is viewed as being "important" if the associated resource availability is the most "scarce".

The steps of the FOX heuristic are the same as those of the TOYODA heuristic except for the calculation of an effective gradient function.

The heuristic methods proposed by Toyoda (TOYODA, 1975), Senju and Toyoda (S – T, 1968), Loulou and Michaelides (L – M M1, M2, SW1, SW2, 1979), Fox and Scudder (FOX, 1985), and Kochenberger *et al.* (KOCHEN, 1974) are analyzed in this chapter.

3.2.3 Effect of Problem Characteristics on Solution Procedure Performance

Study of the Effects of Constraint Slackness

Greedy heuristics for MKPs usually find different solutions based on the problem characteristics. Zanakis (1977) compared the performance of three heuristic methods (Senju – Toyoda (1968), Kochenberger *et al*. (1974), and Hillier (1969)). For the comparison, he created a set of randomly generated 0 – 1 test problems with nonnegative coefficients, and also used benchmark test problems. Zanakis controlled the number of variables (*V*), (20, 60 ,100, 200, 500 and 1000), the number of constraints (*C*), (20, 60, 100 and 1000), and the degree of constraint slackness (30%, 50%, and 90%: the ratio,

$S_i = \frac{b_i}{\sum_{j=1}^{n} a_{ij}}$, expressed as a percentage, is the slackness of the constraint) in the randomly generated test problems. However, each constraint in a given problem had the same degree of constraint slackness. He measured computer running time, error with respect to the best heuristic result, and relative error with respect to the true optimum of the test problems. Zanakis' results suggest all three methods have solution times that increase linearly up to 40-50 variables and 200 constraints, exponentially thereafter, with solution times increasing faster with the number of variables (*V*) vice the number of constraints (*C*). Hillier's (1969) heuristic was the most accurate, but it was much slower than the other two heuristics because of its use of the simplex algorithm. The KOCHEN heuristic was the fastest and the most accurate with both tight (30%) and loose (90%) constraints. In general, the S – T heuristic was the fastest but least accurate on small and medium size problems. Therefore, Zanakis suggested selecting the best heuristics based on the problem characteristics. Loulou and Michaelides (1979) made a similar suggestion based on their research results. Few research efforts have, however, examined or considered these suggestions, an exception being this work and Cho *et al*. (2003b).

Fox and Nachtsheim (1990) evaluated greedy selection rules for 0 – 1 MKPs on their randomly generated test problems. They used six rules (numbered I through VI) to calculate the penalty factor (w_i). The effective gradient (G_j) is $G_j = \frac{c_j}{\sum w_i a_{ij}}$ where w_i represents a "weight" assigned to constraint *i* reflecting its "importance" or the relative "scarcity of resource *i* ", c_j is the *j*th objective function coefficient, and a_{ij} are the constraint coefficients of variable x_j. A greedy rule selects the largest G_j and sets x_j to 1.

Rules I, II and III are based on the Fox and Scudder (1985) heuristic. The method for calculating the effective gradient, G_j, in Rule I is the same as in Equation (15) (Fox and Scudder, 1985). Rule I is modified to create Rules II and III. A constraint tightness consideration is used by these rules. Let s_i^p be the tightness of constraint *i* prior to the *p*th iteration. In Rule I, $s_i^p = b_i^p$, in Rule II where b_i^p is the amount of resource *i* remaining at the end of iteration *p*, $s_i^p = \frac{b_i^p}{\sum_{j \in SC} a_{ij}}$ where *SC* is the set of indices of variables equal to zero, and in Rule III, $s_i^p = b_i^p - \sum_{j \in SC} a_{ij}$.

The weights used are as follows:

$$w_i^p = \begin{cases} 1, & \text{if } \; s_i^p = s_{\min}^p \\ 0, & \text{Otherwise} \end{cases} \tag{17}$$

for all $i \in CT$, where $s_{\min}^p = \min_{i \in CT} \{s_i^p\}$ and *CT* is the set of indices of constraints that could potentially become violated if some of the variables with indices in *V* are set to one.

Rule IV simply sets w_i^p to one for all constraints and iterations. Rules V and VI are based on Toyoda (1975), with weights as follows:

$$w_i^p = \max\left\{ 0, \; \sum_{j \in S} \left(a_{ij} - b_i Q / (b_i)^2 \right) \right\} \tag{18}$$

where *S* is the set of indices of variables set to 1. In Rule V, $Q = 0$, and in Rule VI, $Q = 0.5 \max_{i \in CT} \left\{ \sum_{j \in S} a_{ij} \right\}$.

For their empirical analysis, Fox and Nachtsheim (1990) varied four parameters to randomly generate 1440 test problems. Their parameters were number of variables, number of constraints, constraint matrix density, and slackness ratios (tightness slackness ratio, S_i = 0.3 with probability 0.5 and loose slackness ratio, S_i = 0.7 with probability 0.5). For constraint matrix density, if the constraint matrix contains enough zero entries to be worth taking advantage of them, the constraint matrix may be represented as a sparse matrix. Fox and Nachtsheim measured the average relative efficiencies and the average rank of the objective function among all six rules. They suggested that Rule IV seems to be the best approach in terms of relative efficiencies and rank of the objective function. However, in the mixed slackness ratio problems, Rule I from Fox and Scudder (1985) was superior because Rule IV uses the same importance and same weight for each constraint. Fox and Nachtsheim concluded, “The simplest rule is the best, except when the constraints exhibit mixed slackness.” (Fox and Nachtsheim, 1990)

Studies into Effects of Correlation

Some MKPs can be quickly solved even if *n* is very large, while other problems cannot be easily solved for *n* equal to a few hundred. One reason may be that the correlation between objective function coefficients and each set of constraint coefficients, and the correlation between sets of constraint coefficients effects solution procedure performance. Many authors develop their randomly generated problem sets to verify their algorithm, but few have actually studied the effects of correlation among the test problem coefficients.

Martello and Toth (1988) conducted experiments with an exact algorithm, MT2, solving 0 – 1 knapsack problems with three correlation levels and more than 100,000 variables. They reported uncorrelated and weakly correlated instances were easily solved. However, the strongly correlated instances were very difficult to solve; they could be solved for a small number of variables and constraints, using a dynamic programming algorithm that, however, would not work on larger instances due to excessive space and time requirements. In short, the results of Martello and Toth for the 0 – 1 knapsack problem indicate that problems with near perfect positive correlation between the objective function coefficients and the constraint coefficients are significantly harder to solve than the uncorrelated problems.

Hill and Reilly (2000) measured how the coefficient correlation structure affects solution performance using randomly-generated sets of two-dimensional knapsack test problems. For test sets, they controlled three problem generation parameters: type of correlation measure (Pearson or Spearman), correlation structure, and the constraint slackness; keeping problem size constant.

For their study, they used CPLEX and Toyoda (1975) as solution methods. Their goal was to investigate how problem structure affects solution procedure performance by either exact algorithm (CPLEX) or heuristic (TOYODA). They measured the number of nodes for CPLEX performance and the relative error for TOYODA performance. Between Pearson and Spearman test problems, Spearman correlation problems were harder to solve. For correlation structure, they found that the difficult problems requiring more CPLEX nodes have larger differences between ρ_{CA^1}, ρ_{CA^2}, and $\rho_{A^1A^2}$ where ρ_{CA^i} $(i = 1, 2)$ is the correlation between objective function coefficients and the *i*th constraint coefficients and $\rho_{A^1A^2}$ is the correlation between the first constraint coefficients and second constraint coefficients. Negative values of $\rho_{A^1A^2}$ yield the hardest problems for TOYODA. Interestingly, the challenging problems for CPLEX were easy for TOYODA as TOYODA often found optimal solutions. For constraint slackness, tight constraints provide more challenging problems for both CPLEX and TOYODA. The interaction between correlation structure and constraint slackness is that tighter constraints and constraint coefficients with a wider range of values produce more difficult problems. However, in CPLEX, positive interconstraint correlation usually yields easy to solve problems. For TOYODA, tight constraints and negative $\rho_{A^1A^2}$ make problems harder to solve. Their results indicate that an algorithms' performance depends on the problem "characteristics".

3.2.4 Problem Generation

Zanakis (1977) examined the performance of three heuristics on randomly generated test problems. He suggests that number of variables, number of constraints, and constraint slackness ratios are considerable factors that affect the performance of heuristics. The degree of constraint slackness was studied at three levels. For each combination, c_j and a_{ij} are generated from a uniform distribution between 0 and 40, and then the RHS, b_i, is set to equal to $S \cdot \sum_{j=1}^{n} a_{ij}$ where S is the fixed slackness ratio.

Fox and Nachtsheim (1990) follow the guidelines of the experimental design from Lin and Rardin (1980) to develop a generalized packing problem set for comparing six different algorithms. They define (1) number of constraints, m, (2) number of variables, n, (3) constraint matrix density, d, with $a_{ij} = 0$ according to the density, and (4) constraint slackness ratios, s. Two different levels of m, n, d, and three different levels of s are recommended in the randomly generated problem set. The values are as follows: m = 50 or 200, n = 50 or 200, and d = 0.05 or 0.10 (5 % of the elements of the constraint matrix are not zero or 10% are not zero). Although they used the same method to create constraint slackness ratios as Zanakis (1977), they introduced mixed slackness ratios, where S_i = 0.30 with probability 0.5 and S_i = 0.70 with probability 0.5. They suggest the use of a split plot or repeated measure design with 2×2×2×3 ($m \times n \times d \times s$) factorial structure, and the response is the objective function value.

Beasley (2004) provides a MKP benchmark test problem set (from Chu and Beasley (1998)). Most all MKP studies employ these problems as their benchmark. The data sets, labeled *mkbapcb1* through *mknapcb9*, are synthetically generated problem sets with 30 problems in each file. The test problems use the convention of equal slackness ratios in all constraints of a particular problem. The ratio is varied across three constraint slackness ratios (0.25, 0.5, 0.75). Within a problem set file, each ratio is applied to 10 problems. The test sets involve problems with 5, 10, and 30 constraints and 100, 200, and 500 variables. The characteristics of these test problem sets are discussed in the next section.

3.3 The Lack of Diversity of the Beasley Problem Set

Standard problem sets are commonly used to benchmark heuristic performance. One such set, available at Beasley (2004), has long been the favored problem set. In fact, one is hard pressed to find a more common problem set. Benchmark problems have a recognized and valued role in empirical analysis, providing an objective, reusable basis of comparison. Empirical analysis guidelines recognize this purpose and encourage the use of benchmark problems.

However, "standard" problems lack the diversity obtainable via a structured, experimental design-based, synthetic problem generation method. This lack of diversity can lead to false claims of solution method generality. Furthermore, the lack of diversity can lead to oversight error (Hill, 1998). As defined, oversight error implies missing an important factor in an analysis of an algorithm because the researcher was unaware of the factor's relevance and its presence. This definition of oversight error is expanded to include a failure to recognize important algorithm performance attributes due simply to insufficient test problem diversity.

Three common indicators of problem difficulty are the number of variables in the problem, the number of constraints in the problem, and the "tightness" of the constraints. Lack of diversity in the standard problem set is found in two important characteristics: constraint slackness and problem coefficient correlation structure. These characteristics are not sufficiently varied in benchmark problems.

Constraint slackness is defined as the ratio of the value of the right-hand side parameter to the sum of the constraint coefficients. The value, S_i, for $i = 1, \ldots, m$ constraints, is called the slackness of a constraint. For the standard problems examined, S_i=S for all constraints in a particular problem and S was varied with values of 0.25, 0.50 and 0.75. This means all constraints within a given problem have a similar structure with respect to slackness settings.

A portion of a problem's correlation structure is the correlation between the set of objective function coefficients and each set of constraint coefficients. Hill and Reilly (2000) systematically varied these values, denoted as ρ_{CA^i} for $i = 1, 2$ and found that algorithm performance varied as values of ρ_{CA^i} varied. The benchmark problems do not sufficiently vary ρ_{CA^i}. In fact, any variation of ρ_{CA^i} away from a value of approximately 0.42 is due merely to sampling error. Chu and Beasley (1998) induce correlation using a linear function. This linear function yields a theoretical value of $\rho_{CA^i} = 0.42$.

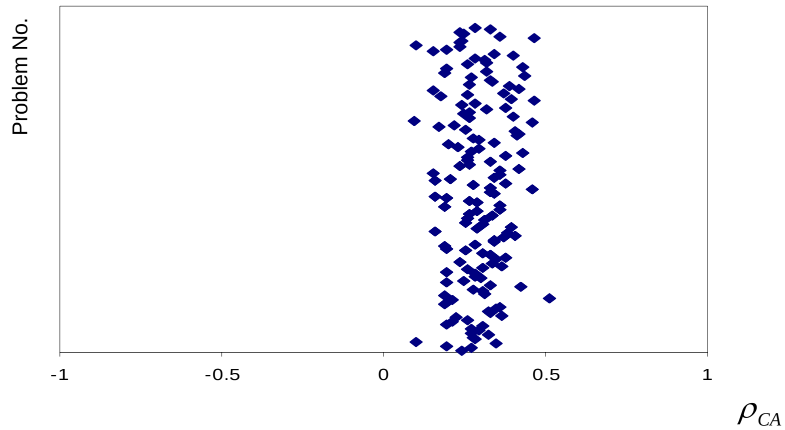
Table 1. Correlation Analysis of Standard Problems

File	min ρ_{CA^i}	max ρ_{CA^i}	(n,m)
mknapcb1	0.094	0.511	(100,5)
mknapcb2	0.163	0.461	(250,5)
mknapcb3	0.189	0.403	(500,5)
mknapcb4	-0.157	0.459	(100,10)
mknapcb5	0.003	0.326	(250,10)
mknapcb6	0.030	0.308	(500,10)
mknapcb7	-0.256	0.437	(100,30)
mknapcb8	-0.192	0.307	(250,30)
mknapcb9	-0.074	0.213	(500,30)

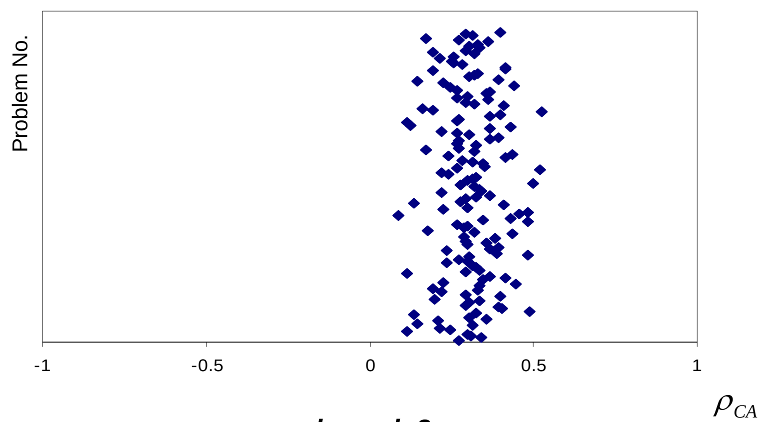
(n,m) represents (variables, constraints) in problems

Table 1 summarizes the analysis of these problems. Among the nine files available at Beasley (2004) containing the 270 test problems, *mknapcb1.txt* through *mknapcb9.txt*, each ρ_{CA^i} varies randomly and provides an insufficient range. Not surprisingly, the maximum ρ_{CA^i} gets smaller as the problem size increases.

Figure 5 shows the correlation range between objective function coefficient and five constraint coefficients in *mknapcb1* and *mknapcb2*. All correlation values (ρ_{CA^i} , i = 1, …, 5) between objective function coefficients and each set of constraint coefficients fall in the range from 0 to 0.5. This range of correlation values is very narrow when compared to the entire correlation range -1 to 1.



mknapcb 1



mknapcb 2

Figure 5. Correlation Range of ρ_{CA^i} in *mknapcb1* and *mknapcb2*

Two key insights arise from this analysis. The range of ρ_{CA^i} is limited in each problem set thereby limiting algorithm performance insight at stronger, particularly negative, levels of coefficient correlation. These problems do not provide an experimental basis upon which to make claims of general applicability of a particular solution procedure. The second insight is that some of the randomly attained values of ρ_{CA^i} are likely strong enough to exert an influence on algorithm performance; an impact rarely, if ever, accounted for in empirical analyses (*i.e.*, potential oversight error).

Hill and Reilly (2000) determined that varied levels of constraint tightness and problem correlation structure within a problem have a significant influence on heuristic (and enumerative) procedure performance. Thus, the original Beasley problem sets were modified slightly in order to show how the varied levels of constraint tightness and correlation structure affect the performance of heuristics. The five legacy heuristics, TOYODA, S – T, L – M M1, FOX and KOCHEN were used to solve three different versions of the *mknapcb1* problems: original, modified constraint slackness, and modified correlation problem sets. The original problem set is the unmodified *mknapcb1* test set. In the modified constraint slackness problem, constraint slackness settings are changed in the original problem sets so that one constraint slackness setting is tight, with a slackness ratio of 0.3, and the other constraint slackness settings are loose with a slackness ratio of 0.7. This changes the basic structure of the test problem. In the modified correlation problem, the order of constraint coefficients in one of the original constraints is shuffled to match the order of the objective function coefficients, thus one of the correlation values (between the coefficients of the objective function and one of the constraints) is strongly positive, while the other objective function – constraint correlations are kept at their original, weak positive, correlation values. The results for each problem set are provided in Table 2. To compare heuristics, the number of times each heuristic returned the clear best solution is counted, compared to all other heuristics considered (ties are excluded).

Table 2. Results of Greedy Heuristics on *mknapcb1* Problem Set

	Original Problem	Modified Slackness	Modified Correlation
TOYODA	3	0	1
S – T	0	27	3
L – M M1	0	0	11
FOX	0	0	8
KOCHEN	23	1	3

Key:

Modified Slackness	One constraint's slackness is tight and others are loose (tight = 0.3, loose = 0.7)
Modified Correlation	One correlation coefficient is close to one (Highly positive)
TOYODA	Toyoda (1975)
S – T	Senju and Toyoda (1968)
L – M M1	Loulou and Michaelides Method 1 (1979)
FOX	Fox and Scudder (1985)
KOCHEN	Kochenberger *et al*. (1974)

The results in Table 2 are surprising! KOCHEN is a clear winner for the original *mknapcb1* problem set. S – T, however, is the best heuristic when just one constraint is tightened. If one of the correlation values between the objective function coefficients and constraint coefficients, ρ_{CA^i} , $i = 1, \ldots, 5$, is close to one, L – M M1 or FOX became the best performing heuristic among the five heuristics examined. Therefore, simple perturbations to Beasley's standard benchmark MKP set provide quite different problem characteristics and result in different heuristics performing well. Such changes, simple as they are, mean a heuristic cannot be clearly identified as a best heuristic based on this benchmark problem test set. Therefore, empirical analysis of heuristics requires a more diverse test set that includes various problem characteristics.

3.4 Approaches to the Empirical Analysis of Heuristics

3.4.1 General Approach for Empirical Testing of Heuristics

Barr *et al*. (1995) outline a general approach for conducting empirical testing of heuristics. Their guidelines, and the specifics implemented, are as follows:

1. Define the goals of the experiment:

The goal in this research is to conduct a rigorous computational study to isolate and examine the performance of eight greedy heuristics: TOYODA, S – T, the four L – M (M1, M2, SW1, SW2), FOX and KOCHEN, based on constraint slackness and correlation structure. Specifically, one purpose of this research is to gain insight into how constraint slackness and correlation structure affect the performance of different heuristics, and develop new heuristics based on insights gained. In other words, which heuristic method yields the best solution under certain correlation and slackness conditions, why does this happen, and how might this knowledge be used to create better heuristics?

2. Choose measures of performance and factors to explore:

A factor is any controllable variable in an experiment that affects the outcome of the experiment. The factors in this experiment are the constraint slackness and correlation structure.

A measure is the outcome of an experiment. The measure is the value of the objective function and relative error. This is used to determine which heuristic performs best for each combination of the levels of the factors.

3. Design and execute the experiment:

As mentioned in Section 3.2, the experiment uses problems involving a full range of constraint slackness and feasible correlation structures. Two types of problems are used for the experiments: 2KP and 5KP test problems.

A 2KP test problem set from Hill and Reilly (2000) has 1120 test problems. In Section 3.5.1, the characteristics of the problems in this test set are described.

The 5KP set is developed in a manner similar to that used by Hill and Reilly (2000). However, Section 3.6.1 discusses the difficulty in generating 5KP test problems versus 2KP test problems. Each problem in the 5KP set has 100 variables and 5 constraints. The greedy heuristics have a polynomial computation time (Akcay *et al*., 2002) because greedy heuristics select an item at each iteration and the total number of iterations is less than or equal to the number of variables. As the number of variables in the problem increases, solution time increases linearly. Since the focus of the research is the performance of a heuristic based on constraint slackness and correlation structure, the number of variables is fixed to 100. This removes performance variation due to problem size.

4. Analyze the data and draw conclusions:

Barr *et al*. (1995) suggest that there are at least three sources of variation one must recognize. These are as follows: (1) variation among the performance of the algorithms, (2) variation due to problem parameters, and (3) variation within problems. The research focuses on (1) and (2), with emphasis on (1), to gain new insight into why certain heuristics do well on certain problems. With the new insights gained via empirical testing, the conjecture is that the new heuristics should perform better than other greedy heuristics.

3.4.2 Design of Experiment for Problem Generation

One purpose of this research is to understand heuristic performance based on insights gained from computational tests. Thus, the computational experiments involve running several heuristics on a chosen population of instances and measuring responses. Rardin and Uzsoy (2001) suggest that in order to construct an experimental design, one starts with a set of questions to answer about the heuristics under study: for instance, how different problem characteristics (*e.g.*, problem size, number and nature of constraints), and algorithm components or parameters (*e.g.*, stopping criteria, search neighborhoods and move selection) affect the performance of the heuristics being tested.

Any MKP instance may be characterized by the number of variables, number of constraints, distribution of variables, density of problem matrix, constraint slackness, parameter distribution, and correlation structure. For 2KP, the test instances of Hill and Reilly (2000), used in Cho *et al*. (2003b), consider various constraint slackness and correlation structures with the number of variables and constraints fixed. The characteristics of these 2KP problems are described in section 3.5.1

A full factorial design is a design containing every possible combination of the factor levels. A focus on constraint slackness and correlation structure means constraint slackness and correlation structure are the factors, and their different values are the levels. Only correlation structures that yield a positive semi-definite correlation matrix are used.

Table 3 presents a sample of the 2KP computational experiment matrix. A representative sample of test problem settings and heuristic results are shown. Rows correspond to test instances, while columns show problem characteristics, and columns shaded gray present the objective function values found by the heuristics tested.

Unlike the benchmark problems in the OR library (Beasley (2004)), which are too specific and do not cover all problem characteristics, this randomly generated problem set is general and robust, providing a better measure of the performance of the heuristics.

Table 3. Test Instances vs Heuristics Design for 2KP: A Sample

Prob Num	Rep Num	C1 Slack	C2 Slack	CA1	CA2	A1A2	Best IP	TOYODA	S – T	L – M M1
1	1	1	1	2	2	2	1480	1468	1468	1468
1	2	1	1	2	2	2	1644	1631	1614	1584
1	3	1	1	2	2	2	1497	1458	1450	1441
1	4	1	1	2	2	2	1704	1696	1696	1610
1	5	1	1	2	2	2	1619	1601	1601	1549
2	1	1	2	2	2	2	1647	1626	1635	1615
2	2	1	2	2	2	2	1787	1772	1770	1594
2	3	1	2	2	2	2	1590	1586	1587	1552
2	4	1	2	2	2	2	1629	1627	1628	1464
2	5	1	2	2	2	2	1669	1663	1665	1576
3	1	2	1	2	2	2	1732	1728	1731	1647

Prob Num := problem number
Rep Num := replication number
C1 Slack := slackness of first constraint (1:= 0.3, 2:= 0.7)
C2 Slack := slackness of second constraint (1:= 0.3, 2:= 0.7)
CA1 := correlation between c_j and a_{1j}
(–2:= –0.99997, –1:= –0.49999, 0:= 0, 1:= 0.49999, 2:= 0.99997)
CA2 := correlation between c_j and a_{2j}
(–2:= –0.99773, –1:= –0.49887, 0:= 0, 1:= 0.49887, 2:= 0.99773)
A1A2 := correlation between a_{1j} and a_{2j}
(–2:= –0.99752, –1:= –0.49876, 0:= 0, 1:= 0.49876, 2:= 0.99752)
Best IP := Integer optimal solution
TOYODA := solution by Toyoda's heuristic
S – T := solution by Senju – Toyoda's heuristic
L – M M1:= solution by Loulou – Michaelides' M1 heuristic

3.4.3 Statistical Methods for Analyzing Results

Relative Error

The ultimate goal of a heuristic is to find an optimal solution. Short of this, the heuristic should find a solution close to the optimal solution. If the true optimum, or best known solution, is known for each test problem, a relative error measure based on attained objective function value can be used; that is, a small relative error indicates a solution close to the optimum. The 2KP problems from Hill and Reilly (2000) include the optimal solutions as achieved by CPLEX software, and re-solved using Xpress. Let Z_i be the value of the objective function obtained by heuristic i where i = TOYODA, S – T, the four L – M, FOX, or KOCHEN, and let Z^* be the optimal objective function value. Then the relative error is

$$RE_i = \frac{100 \cdot (Z^* - Z_i)}{Z^*}. \tag{19}$$

Based on the relative error, the best method excluding ties under certain correlation structures and constraint slackness is chosen. Since the goal is to understand why certain correlation and constraint slackness levels make specific methods perform well (*i.e.*, the smallest relative error), the number of times a heuristic is the best, excluding the number of ties, by correlation structure and constraint slackness settings is counted. Relative error results are shown in Table 4. Trends in these counts guide the search for insights. If average relative errors are used to find a best heuristic, it is not proper to apply classical statistical methods like ANOVA and t – tests because the data would not satisfy the assumptions of classical analysis of variance. Classical statistical methods assume that data follows a normal distribution and has constant variance.

However, the variance of the relative errors is very small (near 0), and the distribution is skewed, so the normality assumption is impractical. A Chi-square (χ^2) test and a sign test (nonparametric statistic) that involve counting the number of times a heuristic is best are alternatives to statistically distinguish which heuristic is the best heuristic. Both statistics forgo the traditional assumption that underlying populations are normal. To increase the power of distinction of the best heuristic, the number of ties is excluded. Applicable Chi-square (χ^2) test and sign test are introduced in the next sections.

Table 4. Relative Error of Five Heuristics: A Sample

Problem/ Rep No.	TOYODA	S – T	L – M M1	FOX	KOCHEN	Best Method
1/1	0.8108	0.8108	0.8108	0.4054	0.6757	FOX
1/2	0.7908	1.8248	3.6496	1.2774	0.6691	KOCHEN
1/3	2.6052	3.1396	3.7408	1.8036	0.7348	KOCHEN
1/4	0.4695	0.4695	5.5164	1.5845	0.4695	TIE
1/5	1.1118	1.1118	4.3237	0.4324	0.4324	TIE
2/1	1.2750	0.7286	1.9429	0.7286	0.9107	TIE
2/2	0.8394	0.9513	10.8002	0.0000	0.9513	FOX
2/3	0.2516	0.1887	2.3899	0.9434	0.1887	TIE

Chi-Square Test

A goal of the research is to determine a best method among the greedy heuristics by each correlation structure and constraint slackness. A Chi-square (χ^2) test is used to determine whether or not differences exist among the heuristics. However, this test does not indicate which heuristic is better than the others. Thus, when differences are found to exist, sign tests are then used to determine if a heuristic differs significantly from the others. The χ^2 hypothesis test is:

$$\mathbf{H_0^C} : \text{Heuristic performances do not differ.}$$
$$\mathbf{H_1^C} : \text{At least one heuristic's performance differs from others.} \tag{20}$$

The Chi-square (χ^2) test requires compiling data into bins of equal intervals. For this research, these bins correspond to each heuristic and the counts in each bin are the number of best solutions for the associated heuristic in the problem set (or subset) of interest where j = 1, 2, 3, 4, 5, 6, 7, 8 corresponding to TOYODA, S – T, L – M (M1, M2, SW1, SW2), FOX, and KOCHEN, respectively,

N_j = number of times heuristic j finds unique best solution, for j =1, …, 8 (21)

Next, the expected proportion, p_j, of the N_j that should fall in the jth bin if all bins are equally likely is computed. Since, under $\mathbf{H_0^C}$, it is assumed that there is no difference in the eight methods, p_j= 1/8 for j = 1, 2, …, 8. Finally, the test statistic is

$$\chi^2 = \sum_{j=1}^{8} \frac{(N_j - np_j)^2}{np_j} \tag{22}$$

where n is the number of test problems.

All N_j are equal to or exceed 5, thus χ^2 possess approximately a Chi-square (χ^2) test probability distribution with n – 1 degree of freedom. Since np_j is the expected number of times the jth heuristic is best, if $\mathbf{H_0^C}$ is true, χ^2 is expected to be small. $\mathbf{H_0^C}$ is rejected if χ^2 is too large. For this test, α = 0.1 is used, so the critical value is $\chi^2_{0.1,d.f.}$ where the degree of freedom ($d.f.$) is 7.

Sign Test for Paired Comparison

When the Chi-square test indicates that there are differences among heuristics, a sign test is used to determine whether or not one heuristic outperforms another heuristic. For a sign test, the hypothesis is:

$$\begin{aligned} &\mathbf{H_0^S}: \text{Two heuristics statistically have the same performance.} \\ &\mathbf{H_1^S}: \text{One heuristic has statistically better performance compared to another heuristic.} \end{aligned} \tag{23}$$

If $\mathbf{H_0^S}$ is true, then for any test problem, each heuristic has an equal chance of being the best. Therefore, the distribution of outcomes has the Binomial distribution $B(N, 0.5)$. Let U be the number of times the first heuristic is best. If $\mathbf{H_0^S}$ is true, then $U \sim B(N, 0.5)$ and is approximated by a normal distribution having mean $\mu = N \times \frac{1}{2}$ and standard deviation $\sigma = \sqrt{N \times \frac{1}{2} \times \frac{1}{2}}$. To find the significance level of the result, the following is calculated:

$$P(X \geq U) \approx P(X > U - 0.5) = P\left(Z > \left| \frac{U - 0.5 - \mu}{\sigma} \right| \right) \tag{24}$$

Since the sign test, for $\mathbf{H_1^S}$, shows a best heuristic's perfomance is better than the others, this is a one-tailed test. $\alpha = 0.1$ level of significance is used to decide whether to fail to reject $\mathbf{H_0^S}$ or reject $\mathbf{H_0^S}$.

3.5 Empirical Analyses Based on 2KP

This section discusses results of an empirical study of legacy greedy heuristics using the 2KP test problem set of Hill and Reilly (2000). The purpose of this section is to gain insight into heuristic performance, specifically what causes a heuristic to be the "best" performer (among some set of heuristics).

3.5.1 Test Problem Characteristics in the Library

The Spearman portion of the 2KP problems from Hill and Reilly (2000) is used to examine heuristic performance as a function of constraint slackness and problem correlation structure. For each problem, the number of constraints is 2 (*i.e.*, the 2KP), and the number of variables is 100. These test problems were created using a Spearman rank correlation induction method due to Iman and Conover (1982). This method creates values of trivariate random variables to represent the coefficients (c_j, a_{1j}, a_{2j}) of each variable, and ensures the sets of values have a specified correlation structure. The objective function coefficients, c_j, are integer numbers uniformly distributed from 1 to 100. The coefficients of the first constraint, a_{1j}, are integer numbers uniformly distributed from 1 to 25 while the coefficients of the second constraint, a_{2j}, are integer numbers uniformly distributed from 1 to 40. The three correlation terms are ρ_{CA^1}, ρ_{CA^2}, and $\rho_{A^1A^2}$. The terms ρ_{CA^1} and ρ_{CA^2} represent the correlation between objective function coefficients (c_j) and constraint coefficients (a_{1j} and a_{2j}), respectively. The term $\rho_{A^1A^2}$ represents the correlation between the two constraint coefficients. The correlation levels for each correlation term are set as follows:

$$\rho_{CA^1} \in \{ -0.99997, -0.49999, 0, 0.49999, 0.99997\} \tag{25}$$

$$\rho_{CA^2} \in \{ -0.99773, -0.49887, 0, 0.49887, 0.99773\} \tag{26}$$

$$\rho_{A^1A^2} \in \{ -0.99752, -0.49876, 0, 0.49876, 0.99752\}. \tag{27}$$

These Spearman correlations are calculated by the following formula:

$$\rho_{XY} = 1 - \frac{6\sum_{j=1}^{n}(d_j)^2}{n(n^2-1)} \tag{28}$$

where $d_j = \left| \text{rank of } x_j - \text{ rank of } y_j \right|$ and *n* is the number of variables. The detailed concept of the Spearman correlation is presented in Section 3.5.3.

Within each set of correlation values, the largest absolute values represent the extreme correlation level. Considering each possible combination of correlation value implies 125 combinations. However, of these 125, only 45 represent positive semi-definite correlation matrices which means that the determinant of the correlation matrix must be nonnegative. Rousseeuw and Molenberghs (1994) suggest that, from elementary matrix algebra, it follows that a matrix is a correlation matrix if and only if it is positive semidefinite. For each of these correlation structures, two composite distributions are used: a Type – L composite distribution (minimal independent sampling) and a Type – U composite distribution (maximal independent sampling) (Hill and Reilly, 2000). The joint composite distribution for 11 of these 45 feasible correlations has multiple forms. For the Spearman subset employed, these particular correlation structure settings (11 of the 45 correlation structures) are replicated twice that of the other correlation structures.

For the correlation structure, the five levels of correlation for each correlation term are coded as {-2, -1, 0, 1, 2}, respectively for equations (25) to (27). For example, ρ = (2, 2, 2) indicates ρ_{CA^1} = 0.99997, ρ_{CA^2} = 0.99773, and $\rho_{A^1A^2}$ = 0.99752. Note each correlation level is controlled in the experiment.

For constraint slackness, two different constraint slackness values are examined. A slackness code of 1 represents a slackness value of 0.30 and a slackness code of 2 represents a slackness value of 0.70. The right-hand side coefficients (b_i) are set using the relation:

$$b_i = S_i \sum_{j \in N} a_{ij} \,, \ \ i = 1, 2 \qquad (29)$$

where S_i = 0.30 or 0.70.

Each of the four possible settings of the pair (S_1, S_2) is referred to as a constraint slackness setting. The 1120 problems involve 180 combinations of 45 feasible correlation structures, four-constraint slackness settings, and 5 or 10 replications each (4 × 34 × 5 + 4 × 11 × 10 = 1120: Four constraint slackness settings × feasible correlation structures × replications).

3.5.2 Heuristic Performance Based on 2KP Constraint Slackness

The overall performance of the legacy greedy heuristics is summarized in Table 5. To compare heuristics, the number of times each heuristic returned the best solution compared to all other heuristic solutions (excluding ties) is counted. The purpose of this research is to gain insight into how problem characteristics affect the performance of heuristics. When ties are included in counting, the discrimination power of nonparametric tests used for determining the best heuristics decreases. Even though there are 474 ties when the 1120 problems are heuristically solved, ties are not counted because of the need to distinguish the features of the heuristic which yield the best objective function value. Nevertheless, two statistical tests complement the loss of information by excluding ties: A Chi-square test verifies whether or not a heuristic statistically differs from the others and a sign test indicates which heuristic has better performance compared to the others. The detailed statistical results, Chi-square test and sign test, are provided in Tables A.1 and A.2 of Appendix A: Statistical Tests to Distinguish the Best Heuristic.

Table 5. Number of Times Best by Each Heuristic under Constraint Slackness

Heuristics	(1, 1)	(1, 2)	(2, 1)	(2, 2)
TOYODA	5	8	11	0
S – T	3	57	53	0
L – M M1	2	2	0	4
L – M M2	4	0	1	1
L – M SW1	4	0	0	3
L – M SW2	0	0	0	1
FOX	56	31	37	77
KOCHEN	146	30	28	82
Statistical Tests				
Chi-Square Test	Reject $\mathbf{H}_0^C$	Reject $\mathbf{H}_0^C$	Reject $\mathbf{H}_0^C$	Reject $\mathbf{H}_0^C$
Best by Sign Test	**KOCHEN**	**S – T**	**S – T and KOCHEN**	**KOCHEN**

($\mathbf{H}_0^C$: Heuristic performances do not differ. Reject Region: α =0.1)

The data in Table 5 shows the results of a comparison of one heuristic to all other heuristics simultaneously, while the best heuristics from the sign test are determined by one heuristic compared to each heuristic separately. For example, the slackness setting (2, 2) shows KOCHEN and FOX differing by 5. It means these two heuristics outperform the other heuristics. However, when KOCHEN is compared to just FOX, KOCHEN yields 157 solutions better than FOX's while FOX yields 85 solutions better than KOCHEN's. Thus, the sign test indicates that KOCHEN is the best performer for slackness setting (2, 2). Table 5 overviews which heuristic is the best performer, and statistical tests in Tables A.1 and A.2 confirm whether or not the best heuristic is statistically distinguished.

Analysis of Equal Constraint Slackness, $S_1 = S_2$: (1, 1) or (2, 2)

Two common types of greedy heuristics are primal and dual. Table 5 indicates that KOCHEN is the best heuristic among primal heuristics when both constraints are tight. The dual heuristic, S – T, produces poor performance under constraint slackness settings (1, 1) and (2, 2).

The results suggest two important questions. First, why is KOCHEN the best among primal heuristics? Second, why is a dual heuristic a poor performer for equal constraint slackness values?

First, to understand why KOCHEN outperforms other heuristics for equal constraint slackness values, the penalty cost function of primal heuristics should be studied. All primal heuristics build a penalty function to find a maximum effective gradient based on maximum profit for minimum use of resources. TOYODA is a standard primal effective gradient method. The penalty cost function creates a single effective gradient number based on two limited resources. The penalty cost function was introduced in Equation (10) as follows:

$$v_j = \frac{(P_j \cdot P_u)}{|P_u|} \tag{30}$$

where P_j is the vector (a_{1j}, a_{2j}), P_u is a cumulative total resource used vector, and $|P_u|$ is the norm of P_u. Recall a_{1j} and a_{2j} are coefficients in each constraint. Therefore, v_j depends on the direction of P_u and has no relationship to its magnitude. In words, P_u can be interpreted as weights for each constraint. For example, let $P_u = (w_1, w_2)$ where

$w_1 = \sum_{j \in Selected} a_{1j}$ and $w_2 = \sum_{j \in Selected} a_{2j}$, and *Selected* is the set of selected items. To pick the next item to add, TOYODA calculates an effective gradient as follows:

$$G_j = \frac{c_j}{v_j} = \frac{c_j \cdot |P_u|}{P_j \cdot P_u} = \frac{c_j \cdot \sqrt{(w_1^2 + w_2^2)}}{(a_{1j} \cdot w_1 + a_{2j} \cdot w_2)}. \tag{31}$$

As seen in the above equation, if w_1 is larger than w_2, TOYODA should choose the unselected item with the smallest coefficient, a_{1j}, in the first constraint in order to keep the denominator small and produce the largest G_j. Therefore, if the first constraint has less resource remaining, w_1 is larger than w_2 so TOYODA should give more weight to the first constraint. However, one of TOYODA's important characteristics is a large difference between the value of w_1 and w_2 in the early iterations, while the values of w_1 and w_2 are not very different in the later iterations. The values of w_1 and w_2 in the first iteration are based only on the coefficients, in each constraint, of the first item selected, so the values of w_1 and w_2 for the next selection are exactly the same as the constraint coefficients of the first item picked. Thus, the constraint with the least resource remaining receives more weight than the other constraint in the early iterations. However, for the later iterations, both constraints receive almost equal weights because of the balancing of resource usage by TOYODA. Thus, an item is selected by giving more influence to the most limiting constraint in the early iterations, while an item is selected by considering both constraints in the later iterations.

The algorithm of KOCHEN resembles TOYODA except for the penalty cost function. The penalty cost function of KOCHEN is as follows:

$$v_j = \frac{a_{1j}}{(1 - w_1)} + \frac{a_{2j}}{(1 - w_2)} \tag{32}$$

where $w_i = \sum_{j \in Selected} a_{ij}$, $i = 1, 2$.

The specific characteristic of KOCHEN's penalty cost function magnifies the weight given to the constraint with the least resource remaining during later iterations while TOYODA gives equal weight to each constraint. For slackness setting (1, 1) problems, Figure 6 plots the average weight per iteration variation in v_j for TOYODA (Equation (30)) and KOCHEN (Equation (32)).

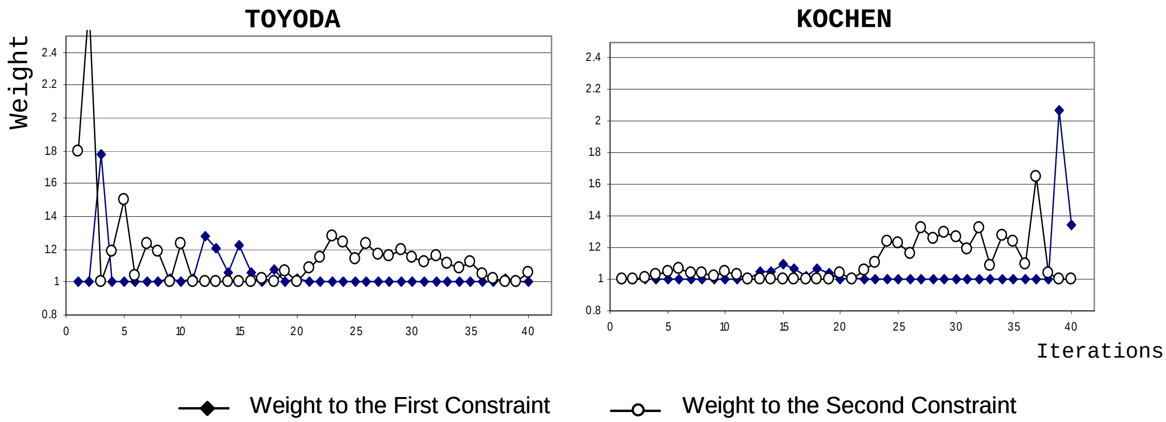


Figure 6. Performance of Weight Trend of TOYODA and KOCHEN for Setting (1, 1)

Each line in each graph represents the weight trend line for each constraint. In the graphs, all weights are scaled to a minimum value of 1. Figure 6 indicates that TOYODA heavily weights constraints early in the solution process, while KOCHEN increases constraint weighting near the end of the solution process. In general, selecting the correct items to add to the knapsacks during early iterations is much easier than during later iterations. Early item selections involve those items with the higher marginal profit (raw profit to resources consumed) making their early selection fairly obvious. As the number of iterations increase, it is more difficult to select a correct item as there is less resource available in each of the constraints and the item choice becomes less obvious (less profits per increased resource cost). Later in the solution process, as each constraint begins to near its limit (the used resource vector approaches the original RHS vector), improved performance equates to improved item selection based on the item's influence on the resource-strapped constraints. The delayed weighting of KOCHEN provides more effective item selection later in the solution process, so KOCHEN achieves better solutions. This allows KOCHEN to more effectively use a scarce resource than TOYODA and to select more items as it uses constraint resources effectively, and this equates directly to an improved objective function value. TOYODA's weighting scheme evens out the influence of any constraint later in the process.

Why the L – M and FOX heuristics find worse solutions than KOCHEN for equal constraint slackness settings is now examined. At each iteration, L – M and FOX consider the most limiting constraint (the constraint with the least resource remaining), while TOYODA and KOCHEN consider all constraints when selecting an item. Thus, FOX and L – M select an item which may have a large c_j, a small a_{ij} in the most limiting constraint, and larger a_{ij} in the other constraints. Larger a_{ij} increase resource usage in the constraints, not including the most limiting constraint, regardless of profit. Therefore, FOX and the four L – M heuristics may not select as many items as other heuristics because of a shortage of resource remaining.

The four L – M penalty cost functions extend the TOYODA penalty cost function. This study examines the four different heuristics created by Loulou and Michaelides (1979). All L – M heuristics (M1, M2, SW1, SW2) use the same penalty cost function, with different parameters. Table 6 indicates that L – M M1 is generally the best performer. Hereafter, only the L – M M1 heuristic is considered.

Table 6. Comparisons of the Four L – M Heuristics

L – M Heuristic	M1 Better	M1 Same	M1 Worse
M2	879	100	141
SW1	254	700	166
SW2	892	91	137

The penalty cost function of L – M M1for the 2KP is as follows:

$$v_j = Max_{i=1,2}\left\{\frac{\left(DA_i + a_{ij}\right)\left(\sum_{k \in SC} a_{ik} - a_{ij}\right)}{1 - DA_i - a_{ij}}\right\}. \qquad (33)$$

As mentioned in section 3.2, when calculating the effective gradient, L – M always picks the higher value for the penalty cost between the two constraints. The amount of resource consumed so far, DA_i, provides the same information as P_u, giving more weight to a constraint with less remaining resource.

FOX uses b_i^p, the amount of constraint *i* remaining instead of P_u. However, b_i^p serves the role of P_u, so FOX gives unit weight to the most limiting constraint, and gives zero weight to the other constraints. In the FOX heuristic, the weight is as follows (see also Equation (16)):

$$w_i = \begin{cases} 1 & \text{if } b_i^p = b_{\min}^p \\ 0 & \text{otherwise} \end{cases} \quad \text{for all } i \in C \qquad (34)$$

where b_i^p is the amount of resource *i* remaining prior to the *p*th iteration, $b_{\min}^p = \min_{i \in C}\left\{b_i^p\right\}$, and *C* represents the set of constraint indices. Thus, FOX selects a new item according to an effective gradient as follows: $G_j = \dfrac{c_j}{\sum_{i \in C} w_i a_{ij}}$. So, FOX considers the constraint with the least remaining resource when selecting an item as does the L – M M1. The difference between L – M M1 and FOX is that L – M M1 also considers the future potential resource remaining after an item selection. Thus, L – M M1 considers more factors than FOX in selecting the most limiting constraint, so L – M M1 should have better performance than FOX in the equal slackness setting with no correlation.

To examine, why S – T is the worst performer for equal constraint slackness setting, Figure 7 plots the average weight values of S – T for constraint slackness setting (1, 1).

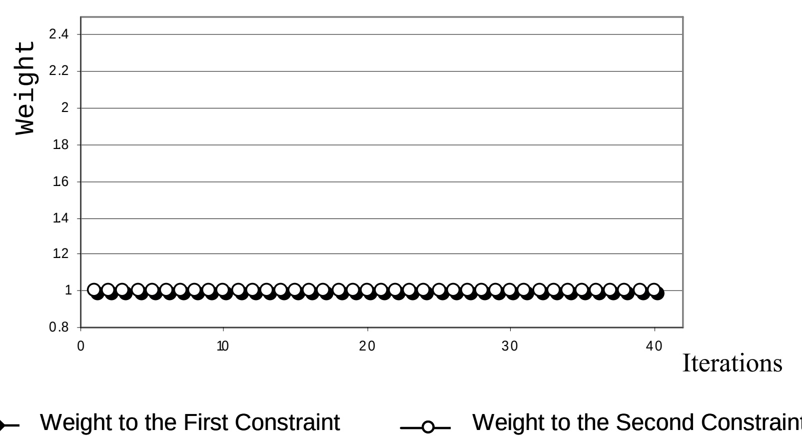


Figure 7. Performance of Weight Trend of S – T for Setting (1,1)

Figure 7 indicates that, when constraint slackness levels are equal, S – T gives the same weight to all constraints using a surplus vector, *R*. This prevents S – T from properly giving weight during later iterations to the constraint with the least resource remaining. The result is ineffective item choices and fewer items put into the knapsack.

Table 7 shows that primal heuristics (TOYODA, KOCHEN, L – M M1 and FOX) yield better solutions in problems with equal constraint slackness settings than the dual heuristic (S – T) because these approaches vary weight vectors for the constraints according to resource usage.

Table 7. Comparison of the Primal Heuristics with the Dual heuristic, S – T, in the Equal Slackness Setting

vs. S – T	Better than S – T	Same to S – T	Worse than S – T
TOYODA	268	284	8
L – M M1	346	20	194
FOX	335	25	200
KOCHEN	429	107	24

During the iterative process, all primal heuristics adapt their weights in order to give more weight to the constraint with the least remaining resource. As a result, the best primal heuristic, KOCHEN, considers all constraints, selects more items, and makes better use of the available resources.

Table 8. Resource Usage in Equal Constraint Slackness Setting by Each Heuristic

Heuristics	Both Const. Tight (1,1)			Both Const. Loose (2,2)		
	# of Vars Selected	Resource Usage 1st Const	Resource Usage 2nd Const	# of Vars Selected	Resource Usage 1st Const	Resource Usage 2nd Const
TOYODA	35.7	0.876	0.896	69.1	0.921	0.928
S – T	34.7	0.842	0.872	68.7	0.914	0.923
KOCHEN	37.6	0.934	0.950	71.4	0.954	0.960
L – M M1	34.7	0.948	0.958	69.1	0.952	0.957
FOX	38.9	0.973	0.978	73.1	0.970	0.979
Optimal	38.1	0.979	0.984	72.0	0.972	0.978

(Ratio of Resource Usage)

Table 8 supports the conclusion that primal heuristics make better use of resources compared to the dual heuristic (S – T). Table 8 also indicates that KOCHEN is more effective than TOYODA and S – T in terms of resource usage. Even though five legacy greedy heuristics (TOYODA, S – T, L – M M1, FOX, and KOCHEN) are analyzed, the heuristics are grouped as [TOYODA, KOCHEN], [S – T], and [FOX, L – M M1]. TOYODA and KOCHEN are primal heuristics that simultaneously consider all constraints with various weights in order to select an item. S – T is a dual heuristic that simultaneously considers all constraints. These three heuristics try to effectively use resources for each constraint. Among these three heuristics, the best heuristic (KOCHEN) uses the most resource for each constraint and selects the largest number of items as shown in Table 8.

FOX and L – M M1 focus on the constraint with the least remaining resource. When these heuristics consider just the tightest constraint, they may pick an item with a large coefficient in the less tight constraint. Thus, FOX and L – M M1 may use more resources regardless of the benefit/cost ratio. Only considering the tightest constraint yields worse solutions and more resource usage when constraint slackness is equal. Table 8 shows that even though FOX selects the same number of items as contained in the optimal solutions, it selects less beneficial items because each iteration considers only the most limiting constraint.

Analysis of Mixed Constraint Slackness, $S_1 \neq S_2$: (1, 2) or (2, 1)

Recall from Table 5 the advantage of S – T when slackness levels are mixed. In Table 5, FOX appears comparable to KOCHEN and S – T on mixed slackness levels. However, the data in Table A.2 of Appendix A compares S – T directly to both KOCHEN and FOX and clearly shows S – T dominance over FOX in the mixed slackness setting and slight S – T preference to KOCHEN.

Senju and Toyoda (1968) suggest that better solutions may be obtained when the right-hand side values differ greatly, but they do not examine this insight further. To understand why S – T performs well in this problem instance, consider problems where $(S_1, S_2) = (0.3, 0.7)$; the first constraint is tight and the second constraint is loose. In the surplus vector of S – T, R can be interpreted as a weight vector: $R = (w_1, w_2)$; w_1 should be larger than w_2 for the slackness setting (0.3, 0.7) because the tight constraint has more unused resource: $w_i = \sum_{j=1}^{n} a_{ij} - b_i$, $i = 1, 2$ (weight in the tight constraint: $w_1 = \sum_{j=1}^{n} a_{1j} - 0.3\sum_{j=1}^{n} a_{1j}$ and weight in the loose constraint: $w_2 = \sum_{j=1}^{n} a_{2j} - 0.7\sum_{j=1}^{n} a_{2j}$). In empirical testing of 2KP, when a problem has mixed constraint slackness settings, the tight constraint is more dominant in determining solution feasibility than the loose constraint. In other words, the loose constraint is typically a non-binding constraint, and has remaining resource at termination. Therefore, a heuristic should give more emphasis to the tight constraint.

Consider the S – T dual effective gradient as follows:

$$G_j = c_j \left(\frac{|R|}{P_j \cdot R} \right) \tag{35}$$

where P_j is the vector (a_{1j}, a_{2j}), c_j is an objective function coefficient, and R is a surplus vector. As shown in Equation (35), S – T uses R to weight both constraints.

Figure 8, adapted from Senju and Toyoda (1968), depicts the surplus vector of the S – T heuristic. The axes represent the sum of coefficients within each constraint, so L_1 and L_2 represent right-hand side values for each constraint for slackness setting (0.3, 0.7). The point T represents the resource usage of the initial, infeasible, point used in the S – T heuristic. The S – T heuristic drops items to force the point T into the feasible region, ideally along vector R.

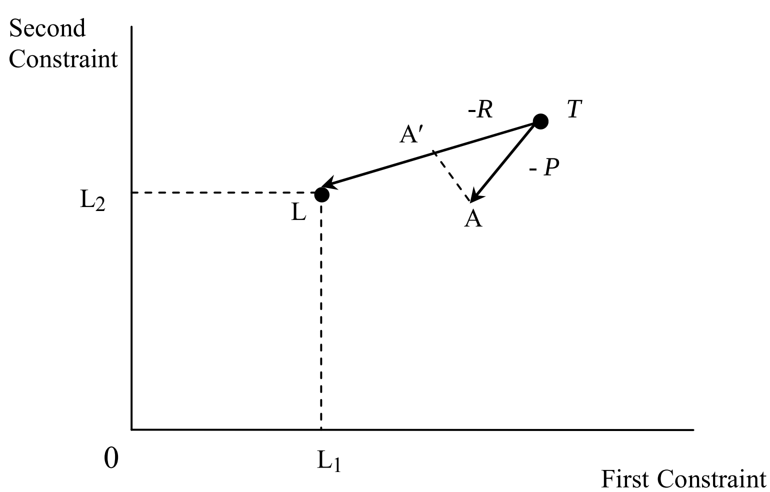


Figure 8. Effective Length of Withdrawal

The slope of *R* is almost horizontal with respect to the first constraint axis, as shown in Figure 8. This implies favoring the first constraint, which in the current case is the tight constraint, over the second constraint. It is preferable to drop items whose objective function coefficients are smaller compared to their projected length on the surplus vector *R*, thus favoring feasibility with respect to the tight constraint. The direction of the surplus vector, *R*, provides the proper direction into the feasible region. In other words, as S – T reaches the point (*L*) of the feasible region, it is using the resources of the two constraints most effectively and favoring the tighter constraint.

For a different view of the surplus vector, *R*, Figure 9 plots the average weight values of the S – T and KOCHEN heuristics for constraint slackness setting (1, 2).

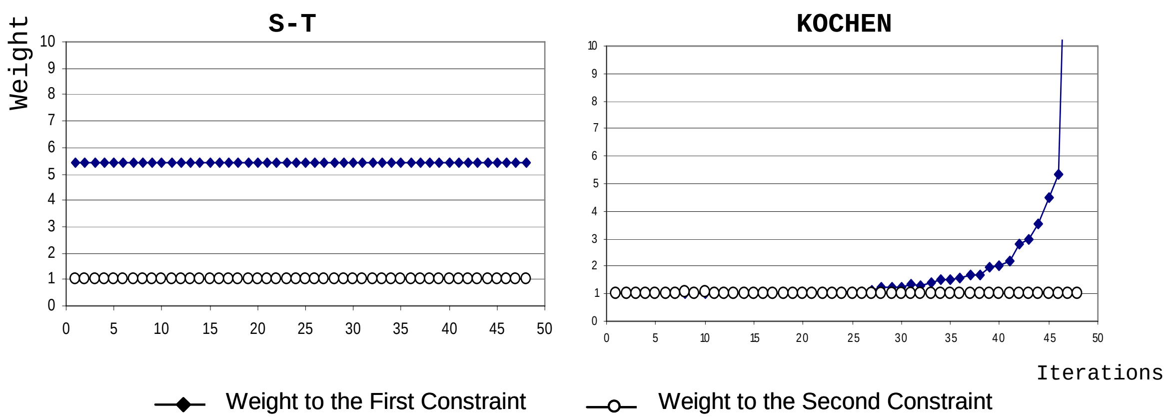


Figure 9. Performance of Weight Trend of S – T and KOCHEN for Setting (1, 2)

S – T focuses item selection with respect to the tight constraint using surplus vector, *R*, during all iterations. As depicted in Figure 9, S – T always gives the tight constraint more weight. KOCHEN also rapidly increases the weight applied to the tight constraint as the number of iterations increases. This significantly increased focus on the tight constraint later in the process allows KOCHEN to achieve results similar to S – T. Thus, S – T and KOCHEN are effective for both (1, 2) and (2, 1) slackness settings because of the emphasis placed on the tight constraint.

In contrast to the S – T dual heuristic and KOCHEN, other primal heuristics select items with weights that balance resource usage. The other primal heuristics do not provide enough emphasis on the tight constraint in the initial iteration. Therefore, the other primal heuristics may select items which have large coefficients in the tight constraint. Table 9 indicates that S – T and KOCHEN select more items than TOYODA, which places nearly equivalent weight on each constraint.

The L – M M1 and FOX heuristics focus on the constraint having the least remaining resource. As shown in Table 9, there is not a large difference in terms of average number of items selected. Even though L – M M1 and FOX always focus on the most limiting constraint, these heuristics choose weaker items during their early iterations because the tight constraint may not be the constraint with the least resource remaining in the early iterations due to initial variable selection.

Even though S – T uses less resources and selects fewer items than KOCHEN and FOX in Table 9, S – T is the best heuristic in Table 5. This emphasizes the importance of picking the best items.

Table 9. Resource Usage in Mixed Constraint Slackness by Each Heuristic

Heuristics	Mixed Const. Slack (1,2)			Mixed Const. Slack (2,1)		
	# of Vars Selected	Resource Usage 1st Const	Resource Usage 2nd Const	# of Vars Selected	Resource Usage 1st Const	Resource Usage 2nd Const
TOYODA	44.1	0.993	0.590	43.5	0.574	0.995
S – T	45.4	0.992	0.630	44.7	0.616	0.995
KOCHEN	45.8	0.994	0.636	45.0	0.621	0.995
L – M M1	41.9	0.994	0.599	41.1	0.583	0.996
FOX	45.9	0.995	0.636	45.1	0.621	0.996
Optimal	45.8	0.998	0.656	45.0	0.640	0.998

(Ratio of Resource Usage)

3.5.3 Heuristic Performance Based on 2KP Correlation Structures

Pearson vs. Spearman Correlation Coefficients

The most commonly used measure for a linear relationship between two variables is the Pearson product-moment correlation coefficient. The two variables must be measured by interval or ratio scale. The values of the correlation can range from -1 to +1. If there is no linear relationship between two variables, the value of their correlation is 0. If there is a perfect positive relationship, the value is +1. If there is a perfect negative relationship, the value is -1. Note that the correlation coefficient measures a linear relationship only. Two variables may have a correlation coefficient close to zero and yet have a very strong nonlinear relationship.

The Spearman (Rank) correlation coefficient is a nonparametric (distribution-free) rank statistic proposed by Spearman in 1904 as a measure of the strength of the association between two variables. If the coefficient is close to positive one, there is a fairly strong positive (linear or nonlinear) relationship between the variables. If the coefficient is close to zero, there is no relationship. These correlation coefficients are based on the ranks of the observations and not on the original data values. In this dissertation, correlation implies Spearman correlation since Hill and Reilly (2000) found that Spearman induced correlation problems were the more difficult to solve as compared to Pearson induced correlation values.

Correlation Influences Heuristic Performance

The relationships among problem coefficients can impact the ability of the effective gradient measures to select the best variable. As Hill and Reilly (2000) note, correlation structure effects algorithm and heuristic performance. Figure 10 shows the relationship between objective function and constraint coefficients according to correlation value.

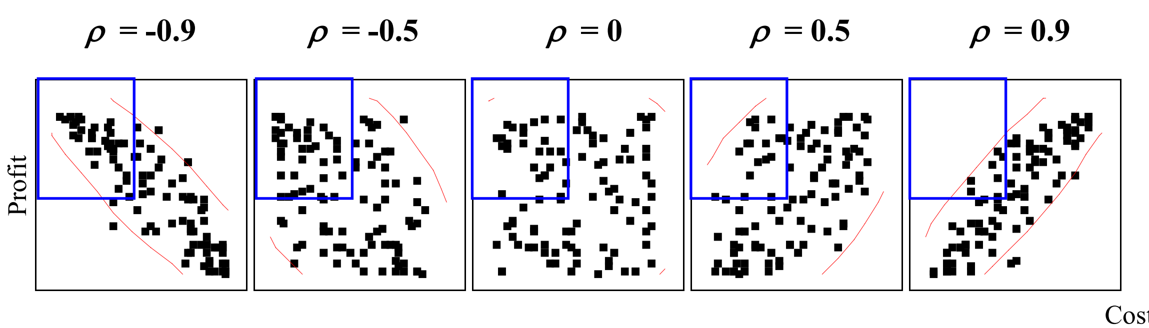


Figure 10. Correlation Graphs According to Correlation Coefficients

Items in the inner square have a relatively large effective gradient value (G_j) so these items are more likely to be selected by a greedy heuristic. Figure 10 indicates that there is a sufficient number of items at $\rho = -0.9$ that are strong candidates for selection by heuristics while, for $\rho = 0.9$, there are few items in the inner square, making it very difficult for heuristics to properly select items to fill the knapsack.

Statement 1. If 1KP has $c_1 > c_2 > ... > c_n$ and $a_1 < a_2 < ... < a_n$ for perfect negative correlation, the Spearman rank correlation coefficient, Equation (28),

$$\rho_{XY} = 1 - \frac{6\sum_{j=1}^{n}(d_j)^2}{n(n^2-1)}$$ gives a value of -1.

Assume that n is an odd number. Figure 11 plots the ranks of c_j and a_j instead of the original values.

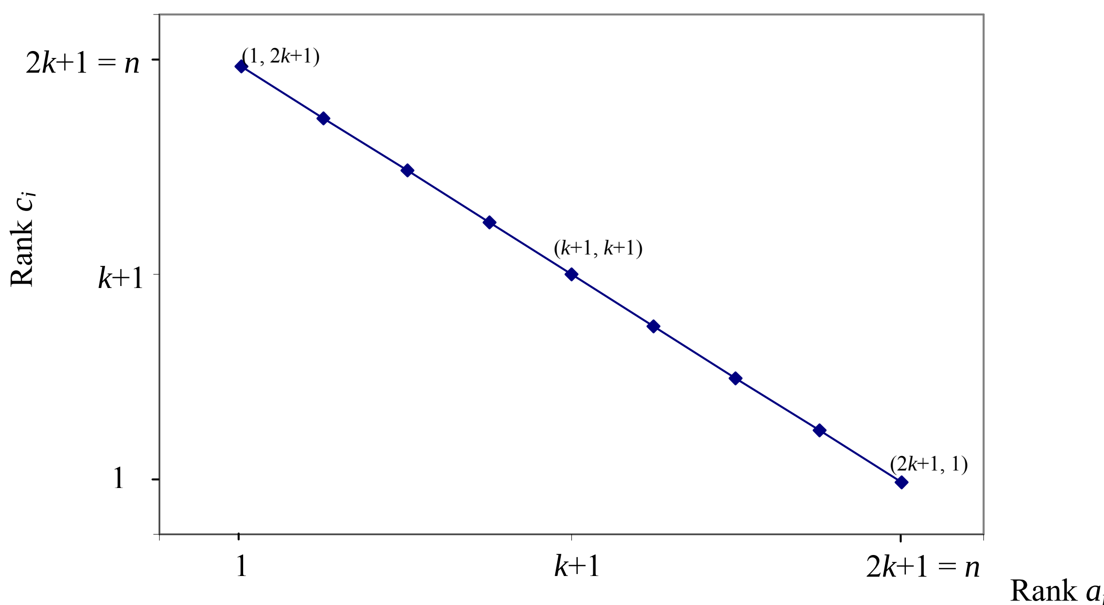


Figure 11. Perfect Negative Correlation Graph

The c_j and a_j ranks become equally spaced. There is a middle point of the line ($k + 1$, $k + 1$) on the line and $n = 2k + 1$.

From Equation (28),

$$\sum_{j=1}^{n} d_j^2 = \sum_{j=1}^{n} \left(\text{rank}\, c_j - \text{rank}\, a_j\right)^2$$

$$= \left((2k+1)-1\right)^2 + \left((2k)-2\right)^2 + \cdots + \left(2-(2k)\right)^2 + \left(1-(2k+1)\right)^2$$

$$= 2 \cdot \left(\left((2k+1)-1\right)^2 + \left((2k)-2\right)^2 + \cdots + \left((k+1)-(k+1)\right)^2\right)$$
$$= 2 \cdot \left((2k)^2 + (2k-2)^2 + (2k-4)^2 + \cdots + 4^4 + 2^2\right)$$

$$= 8 \cdot \left(k^2 + (k-1)^2 + (k-2)^2 + \cdots + 2^2 + 1^2\right)$$

$$= 8 \cdot \sum_{1}^{k} r^2 = \frac{8k(k+1)(2k+1)}{6}$$

and, since $k = \frac{n-1}{2}$, obtain $\sum_{j=1}^{n} d_j^2 = \frac{(n-1)(n+1)n}{3}$.

So, substitute $\sum_{j=1}^{n} d_j^2 = \frac{(n-1)(n+1)n}{3}$ into Equation (28),

$$\rho_{XY} = 1 - \frac{6\sum_{j=1}^{n} (d_j)^2}{n(n^2-1)}$$

$$= 1 - \frac{6\left(\frac{(n-1)(n+1)n}{3}\right)}{n(n^2-1)}$$

$$= 1 - \frac{2(n^2-1)}{(n^2-1)} = 1 - 2 = -1$$

A similar proof holds for even values of *n*. ❑

Theorem 1. Under perfect negative correlation, the greedy heuristic for the zero-one knapsack problem guarantees an optimal solution.

Proof: Assume perfect negative correlation between the objective function coefficients $\left(c_j,\ j = 1, 2, \dots,\ n\right)$ and the constraint coefficients $\left(a_j,\ j = 1, 2, \dots,\ n\right)$. Assume without loss of generality the variables have been re-ordered and re-indexed so that $c_1 > c_2 > \ldots > c_n$. Since perfect negative correlation is assumed, it follows that $a_1 < a_2 < \ldots < a_n$. Let $G_j = \frac{c_j}{a_j},\ j = 1, 2, \ldots, n$ be the effective gradient values for this problem.

Let *A* be a feasible optimal solution to the knapsack problem with *Z*(*A*) the corresponding objective function value. Assume solution *A* is not the greedy heuristic solution to the knapsack problem. Then $\exists\ i,\ j < n$ such that $i < j$ and $x_i = 0$ and $x_j = 1$ in solution *A*. Under the perfect negative correlation assumption, $G_i < G_j$. Create solution *A′* differing from *A* in that $x_i = 1$ and $x_j = 0$ as would be set in a greedy heuristic solution. Since $a_i < a_j$, then *A′* is feasible. Since $c_i > c_j$ then $Z(A) < Z(A')$ which implies *A* is not optimal and therefore a contradiction. Thus, any optimal solution to the single constraint knapsack problem under perfect negative correlation is the greedy heuristic solution to the problem. ❑

Statement 2. Under perfect positive correlation, the greedy heuristic for the zero-one knapsack problem cannot guarantee the optimal solution.

Assume that items are already ordered so that $c_1 < c_2 < ... < c_n$. Since $\rho = 1$ then $a_1 < a_2 < ... < a_n$. Thus, there is a perfect linear relationship defined by an equation of the form: $a_j = k_o c_j$ where k_o is a constant. Then the effective gradient value for the *j*th item is $G_j = \frac{c_j}{a_j} = \frac{c_j}{k_o c_j} = \frac{1}{k_o}$. The effective gradient value for each item is the same: $G_1 = G_2 = \cdots = G_n = \frac{1}{k_o}$. Therefore, the greedy heuristic randomly selects items and may yield a solution worse than the optimal solution. ❑

Positive correlation between objective function coefficients and constraint coefficients make it difficult for a greedy heuristic to select items, while negative correlation values provide a greedy heuristic with obvious items for selection.

In primal heuristics such as the TOYODA, L – M M1, FOX, and KOCHEN to realize a large effective gradient, c_j should be large and v_j should be small, which occurs when ρ_{CA^i} is near extreme negative values, so that item *j* consumes relatively few resources while profit contributed is relatively large. Under these conditions for any ρ_{CA^1} and ρ_{CA^2} near -1, the value of $\rho_{A^1A^2}$ is close to one, meaning resource usage in the constraints are closely matched in the problem. This makes the problems easy for greedy heuristics to find good results. In the 2KP, all legacy heuristics found 18 optimal solutions out of the 20 test problems regardless of constraint slackness as shown in Table 10.

Table 10. Number of Times Optimum Found by Heuristics under Correlation (-2, -2, 2)

Correlation	TOYODA	S – T	L – M M1	FOX	KOCHEN
-2, -2, 2	18	18	18	18	18

Similarly, conditions of weak negative correlation between objective function coefficients and both constraint coefficients, and strong positive constraint coefficient correlation (-1, -1, 2) also possess favorable solution conditions for each heuristic. Under these conditions, the choice of greedy heuristic does not matter since all do well.

Results of Heuristic Solution Performance under Varying Correlation Structures

This section focuses on how the performances of heuristics vary based on problem correlation structures. If mixed constraint slackness is considered, the effect of both varied slackness and correlation structures can be seen. Thus, only equal constraint slackness is considered to focus analysis on the correlation structure effects. Table 11 presents counts of how many times each heuristic yields the best solution (excluding ties) under all 45 correlation structures for problems with equal constraint slackness. Table 12 gives average relative error while Table 13 shows the percentage resource usage of each heuristic's solution and the optimal solution. Table A.2 of Appendix A provides the detailed statistical test to discriminate a best heuristic.

Table 11. Best Performer Counts by Correlation Structure under Equal Slackness in 2KP

Correlation Structure	TOYODA	S – T	L – M M1	FOX	KOCHEN	χ^2 Test Reject $\mathbf{H}_0^c$	Best by Sign Test	Total Probs
2,2,2	0	0	0	2	5	Y	KOCHEN	10
2,1,1	0	0	1	8	1	Y	FOX	10
2,0,0	0	0	1	6	3	Y	FOX	10
2,-1,-1	0	0	0	7	3	Y	FOX	10
2,-2,-2	1	0	0	7	2	Y	FOX	10
1,2,1	0	0	1	9	0	Y	FOX	10
1,1,2	0	0	0	6	1	Y	FOX, KOCHEN	10
1,1,1	0	0	0	0	7	Y	KOCHEN	20
1,1,0	0	0	0	0	6	Y	KOCHEN	10
1,0,1	0	0	0	6	4	Y	FOX, KOCHEN	10
1,0,0	0	0	0	1	19	Y	KOCHEN	20
1,0,-1	0	0	0	0	10	Y	KOCHEN	10
1,-1,0	0	0	0	5	5	Y	FOX, KOCHEN	10
1,-1,-1	0	0	0	3	14	Y	KOCHEN	20
1,-1,-2	0	0	0	0	10	Y	KOCHEN	10
1,-2,-1	0	0	0	2	4	Y	KOCHEN	10
0,2,0	0	0	0	9	1	Y	FOX	10
0,1,1	0	0	0	6	2	Y	FOX	10
0,1,0	0	0	1	1	18	Y	KOCHEN	20
0,1,-1	0	0	0	0	10	Y	KOCHEN	10
0,0,2	0	0	1	1	0	**N**	N/A	10
0,0,1	1	1	2	0	6	Y	KOCHEN	20
0,0,0	0	1	1	1	7	Y	KOCHEN	20
0,0,-1	0	1	0	0	8	Y	KOCHEN	20
0,0,-2	0	0	1	0	2	Y	TOYODA, S - T, KOCHEN	10
0,-1,1	0	0	0	3	4	Y	FOX, KOCHEN	10
0,-1,0	0	0	3	1	12	Y	KOCHEN	20
0,-1,-1	1	0	3	0	4	Y	L-M M1, KOCHEN	10
0,-2,0	0	0	0	4	3	Y	FOX, KOCHEN	10
-1,2,-1	0	0	0	9	1	Y	FOX	10
-1,1,0	0	0	0	3	7	Y	KOCHEN	10
-1,1,-1	0	0	1	3	16	Y	KOCHEN	20
-1,1,-2	0	0	2	0	7	Y	KOCHEN	10
-1,0,1	0	0	0	5	2	Y	FOX, KOCHEN	10
-1,0,0	0	0	3	1	13	Y	KOCHEN	20
-1,0,-1	1	0	2	0	6	Y	KOCHEN	10
-1,-1,2	0	0	1	1	0	**N**	N/A	10
-1,-1,1	0	0	1	4	2	Y	TOYODA, KOCHEN	20
-1,-1,0	1	0	1	0	1	Y	TOYODA, S - T, KOCHEN	10
-1,-2,1	0	0	0	3	4	Y	FOX, KOCHEN	10
-2,2,-2	0	0	0	8	1	Y	FOX	10
-2,1,-1	0	0	0	2	6	Y	KOCHEN	10
-2,0,0	0	0	0	4	4	Y	FOX, KOCHEN	10
-2,-1,1	0	0	0	4	3	Y	L -M M1, FOX, KOCHEN	10
-2,-2,2	0	0	0	0	0	N/A	N/A	10

(Reject Region: α =0.1)

Table 12. Relative Error of Each Heuristic by Correlation Structure (under (1,1) and (2,2))

Correlation Structure	TOYODA	S – T	L – M M1	FOX	KOCHEN
2,2,2	0.765	0.921	3.146	0.700	0.418
2,1,1	3.145	3.997	3.954	0.845	1.638
2,0,0	4.347	5.648	4.414	2.006	1.746
2,-1,-1	4.960	5.914	5.559	1.628	2.061
2,-2,-2	2.439	3.353	2.606	0.787	1.090
1,2,1	3.417	4.343	2.188	0.573	1.910
1,1,2	0.293	0.273	2.476	0.088	0.258
1,1,1	0.525	0.798	3.400	3.456	0.295
1,1,0	0.627	0.907	3.722	8.116	0.244
1,0,1	1.799	2.437	2.967	1.097	0.559
1,0,0	2.059	3.221	2.010	4.264	0.343
1,0,-1	2.521	5.569	2.530	7.232	0.607
1,-1,0	2.877	4.211	2.525	1.680	0.652
1,-1,-1	3.288	5.602	1.549	4.486	0.526
1,-1,-2	3.916	6.662	1.193	9.593	0.496
1,-2,-1	2.449	5.373	2.714	1.176	0.501
0,2,0	3.803	4.905	3.833	1.126	2.098
0,1,1	2.795	3.751	2.820	0.774	0.810
0,1,0	3.310	4.368	2.426	3.402	0.531
0,1,-1	3.651	5.261	2.179	7.351	0.454
0,0,2	0.145	0.145	1.490	0.141	0.100
0,0,1	0.800	0.805	1.936	1.853	0.365
0,0,0	0.469	0.827	1.770	3.492	0.381
0,0,-1	0.660	1.105	1.864	6.476	0.418
0,0,-2	0.437	0.437	0.810	10.380	0.480
0,-1,1	1.073	1.859	1.005	0.795	0.236
0,-1,0	1.724	2.132	1.087	2.199	0.346
0,-1,-1	1.666	2.042	1.002	4.926	0.509
0,-2,0	1.843	2.950	1.192	0.553	0.379
-1,2,-1	4.297	6.275	5.535	0.867	2.060
-1,1,0	2.985	4.439	2.261	1.539	0.447
-1,1,-1	3.983	7.157	1.996	4.580	0.553
-1,1,-2	4.585	9.753	0.910	9.702	0.525
-1,0,1	1.067	1.385	1.282	0.550	0.375
-1,0,0	1.772	2.698	0.806	2.289	0.362
-1,0,-1	1.428	4.087	0.888	4.620	0.408
-1,-1,2	0.104	0.104	0.590	0.069	0.076
-1,-1,1	0.616	0.786	0.732	0.960	0.263
-1,-1,0	0.226	0.570	0.746	2.205	0.279
-1,-2,1	0.667	0.929	0.425	0.365	0.188
-2,2,-2	2.605	3.835	3.254	0.511	1.317
-2,1,-1	2.607	5.637	3.411	1.082	0.399
-2,0,0	1.602	2.825	1.263	0.612	0.257
-2,-1,1	0.577	0.981	0.422	0.385	0.298
-2,-2,2	0.003	0.003	0.011	0.011	0.003

Unit: Percent

Table 13. Resource Usage by Each Heuristic Solution and Optimal Solution (under (1,1) and (2,2))

Correlation Structure	TOYODA		S – T		L – M M1		FOX		KOCHEN		OPTIMUM	
	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd
2,2,2	0.998	0.977	0.998	0.973	0.997	0.981	0.999	0.996	0.998	0.985	1.000	0.997
2,1,1	1.000	0.602	1.000	0.578	1.000	0.876	0.998	0.920	1.000	0.717	1.000	0.933
2,0,0	1.000	0.481	1.000	0.439	1.000	0.763	0.999	0.903	1.000	0.715	1.000	0.898
2,-1,-1	1.000	0.376	1.000	0.357	1.000	0.627	0.998	0.888	1.000	0.663	1.000	0.883
2,-2,-2	1.000	0.298	1.000	0.250	1.000	0.476	0.998	0.751	1.000	0.484	1.000	0.715
1,2,1	0.601	1.000	0.578	1.000	0.872	1.000	0.902	1.000	0.686	1.000	0.905	1.000
1,1,2	0.985	0.991	0.986	0.991	0.995	0.995	0.992	0.995	0.986	0.991	0.994	0.996
1,1,1	0.991	0.991	0.987	0.986	0.997	0.998	0.995	0.999	0.995	0.997	1.000	0.999
1,1,0	0.989	0.993	0.988	0.988	0.997	0.998	0.997	0.998	0.996	0.998	0.998	1.000
1,0,1	0.999	0.868	1.000	0.847	0.999	0.933	0.999	0.972	0.998	0.936	1.000	0.979
1,0,0	1.000	0.901	1.000	0.863	0.997	0.988	0.994	0.992	0.997	0.979	0.999	0.996
1,0,-1	1.000	0.912	0.999	0.838	0.997	0.994	0.995	0.990	0.997	0.984	0.999	0.999
1,-1,0	1.000	0.723	1.000	0.684	0.999	0.886	0.998	0.943	1.000	0.884	0.999	0.943
1,-1,-1	1.000	0.782	1.000	0.714	0.996	0.962	0.993	0.986	0.997	0.948	1.000	0.977
1,-1,-2	1.000	0.816	1.000	0.736	0.994	0.982	0.993	0.987	0.996	0.990	1.000	0.996
1,-2,-1	1.000	0.646	1.000	0.530	0.999	0.747	0.999	0.917	0.999	0.812	1.000	0.881
0,2,0	0.456	1.000	0.424	1.000	0.740	1.000	0.858	0.999	0.597	1.000	0.875	1.000
0,1,1	0.783	1.000	0.758	1.000	0.892	0.999	0.961	1.000	0.890	0.999	0.970	1.000
0,1,0	0.834	1.000	0.804	1.000	0.988	0.997	0.991	0.998	0.971	1.000	0.996	1.000
0,1,-1	0.865	1.000	0.813	1.000	0.993	0.995	0.989	0.996	0.990	0.998	0.998	0.999
0,0,2	0.975	0.990	0.975	0.990	0.985	0.993	0.979	0.991	0.977	0.990	0.981	0.994
0,0,1	0.976	0.988	0.973	0.989	0.995	0.994	0.994	0.996	0.989	0.993	0.998	0.998
0,0,0	0.984	0.992	0.976	0.987	0.993	0.993	0.994	0.993	0.989	0.992	0.998	0.999
0,0,-1	0.981	0.991	0.968	0.993	0.991	0.991	0.990	0.993	0.992	0.994	0.999	0.999
0,0,-2	0.988	0.986	0.987	0.986	0.990	0.992	0.989	0.988	0.986	0.989	0.997	0.997
0,-1,1	0.999	0.866	0.999	0.838	0.996	0.919	0.996	0.960	0.999	0.920	0.999	0.956
0,-1,0	0.998	0.880	0.998	0.872	0.993	0.975	0.992	0.981	0.995	0.962	0.999	0.989
0,-1,-1	0.997	0.926	0.997	0.913	0.992	0.990	0.992	0.984	0.995	0.977	0.999	0.998
0,-2,0	0.999	0.793	0.999	0.744	0.998	0.872	0.996	0.956	0.998	0.913	1.000	0.946
-1,2,-1	0.355	1.000	0.293	1.000	0.539	1.000	0.821	0.999	0.543	1.000	0.814	1.000
-1,1,0	0.691	1.000	0.644	1.000	0.868	0.999	0.940	0.996	0.893	1.000	0.936	1.000
-1,1,-1	0.726	1.000	0.621	1.000	0.936	0.998	0.966	0.997	0.936	0.999	0.980	1.000
-1,1,-2	0.788	1.000	0.636	1.000	0.988	0.992	0.989	0.987	0.979	0.998	0.993	1.000
-1,0,1	0.842	0.999	0.823	0.999	0.895	0.998	0.942	0.993	0.890	0.998	0.946	1.000
-1,0,0	0.878	1.000	0.843	1.000	0.980	0.995	0.981	0.996	0.964	0.998	0.988	0.999
-1,0,-1	0.909	0.999	0.814	1.000	0.988	0.994	0.979	0.993	0.974	0.996	0.991	1.000
-1,-1,2	0.986	0.974	0.986	0.974	0.990	0.978	0.987	0.976	0.987	0.975	0.992	0.979
-1,-1,1	0.956	0.995	0.948	0.995	0.988	0.992	0.981	0.991	0.972	0.995	0.991	0.997
-1,-1,0	0.986	0.996	0.986	0.983	0.992	0.993	0.990	0.992	0.988	0.995	0.997	0.999
-1,-2,1	0.998	0.876	0.999	0.868	0.993	0.939	0.993	0.961	0.995	0.924	0.998	0.963
-2,2,-2	0.313	1.000	0.265	1.000	0.405	1.000	0.714	0.998	0.431	1.000	0.677	1.000
-2,1,-1	0.603	1.000	0.504	1.000	0.702	1.000	0.876	0.999	0.806	1.000	0.884	1.000
-2,0,0	0.724	1.000	0.671	1.000	0.828	0.997	0.892	0.994	0.830	0.999	0.892	1.000
-2,-1,1	0.863	0.998	0.840	0.997	0.913	0.993	0.944	0.994	0.894	0.995	0.932	0.999
-2,-2,2	0.971	0.955	0.971	0.955	0.971	0.954	0.971	0.954	0.971	0.955	0.971	0.957

Hypothesis of Heuristic Solution Performance under Correlation Structures

In an analysis of heuristics based on correlation structures, the goal is to determine whether or not the solution performance of heuristics are affected by correlation structures. The best performing heuristic for slackness is the one that most effectively weights the dominant constraint (tighter constraint). Thus, if the correlation structures create a dominant constraint, as slackness settings do, the best heuristic should thus effectively weight the dominant constraint created by correlation structures. Thus, the hypothesis for correlation structure is as follows:

Hypothesis of 2KP Correlation Structure

When a 2KP possesses equivalent slackness levels, problem correlation structure can dictate a dominant constraint.

Analysis of Hypothesis of 2KP Correlation Structure

The performance of heuristics is affected by various correlation structures as shown in Tables 11 and 12. Constraints possessing positive correlation with the objective function are said to dominate. As previously established, a tight constraint, in a mixed slackness problem, dominates a 2KP problem from a slackness perspective. Heuristics should focus on the dominant constraint(s) when selecting items to place in the knapsack.

KOCHEN and FOX are the best performing heuristics when examining results by correlation structure. Table 13 data display an interesting trend. When ρ_{CA^1} and ρ_{CA^2} values are similar, resource usage values for each constraint are similar. When the values differ, the constraint with the higher positive correlation value has the higher resource usage. This holds even when $\rho_{A^1A^2}$ is negative.

Table 14 shows those correlation structures having similar resource usage in each constraint (difference is less than 3%). Note this occurs in 26 of the 45 different structures and in all cases there is not a large difference between ρ_{CA^1} and ρ_{CA^2}.

Table 14. Correlation Structures with Less Than 3% Difference in Resource Usage by Each Constraint and Best Performing Heuristic

Corr Structure	%Difference in Resource Usage	Best Performing Heuristic	Corr Structure	%Difference in Resource Usage	Best Performing Heuristic	Corr Structure	%Difference in Resource Usage	Best Performing Heuristic
2,2,2	0.3 %	KOCHEN	0,1,0	0.4 %	KOCHEN	-1,1,-1	2.0 %	KOCHEN
1,1,2	0.2 %	FOX, KOCHEN	0,1,-1	0.1 %	KOCHEN	-1,1,-2	0.7 %	KOCHEN
1,1,1	0.1 %	KOCHEN	0,0,2	1.3 %	KOCHEN	-1,0,0	1.1 %	KOCHEN
1,1,0	0.2 %	KOCHEN	0,0,1	0.0 %	KOCHEN	-1,0,-1	0.9 %	KOCHEN
1,0,1	2.1 %	FOX, KOCHEN	0,0,0	0.1 %	KOCHEN	-1,-1,2	1.3 %	All Heuristics
1,0,0	0.3 %	KOCHEN	0,0,-1	0.0 %	KOCHEN	-1,-1,1	0.6 %	TOYODA, KOCHEN
1,0,-1	0.0 %	KOCHEN	0,0,-2	0.0 %	TOYODA, S-T, KOCHEN	-1,-1,0	0.2 %	TOYODA, S-T, KOCHEN
1,-1,-1	2.3 %	KOCHEN	0,-1,0	1.0 %	KOCHEN	-2,-2,2	1.4 %	All Heuristics
1,-1,-2	0.4 %	KOCHEN	0,-1,-1	0.1 %	KOCHEN			

When constraints have similar values for ρ_{CA^1} and ρ_{CA^2}, a greedy heuristic must balance weight on these constraints in order to use resources evenly in early iterations and then, in the later iterations, assign greater weight to the constraint with the least resource remaining. KOCHEN's delayed weighting scheme accomplishes this. Figure 12 shows the average performance of weight variation on each constraint of TOYODA, S – T, and KOCHEN for correlation structure (2, 2, 2).

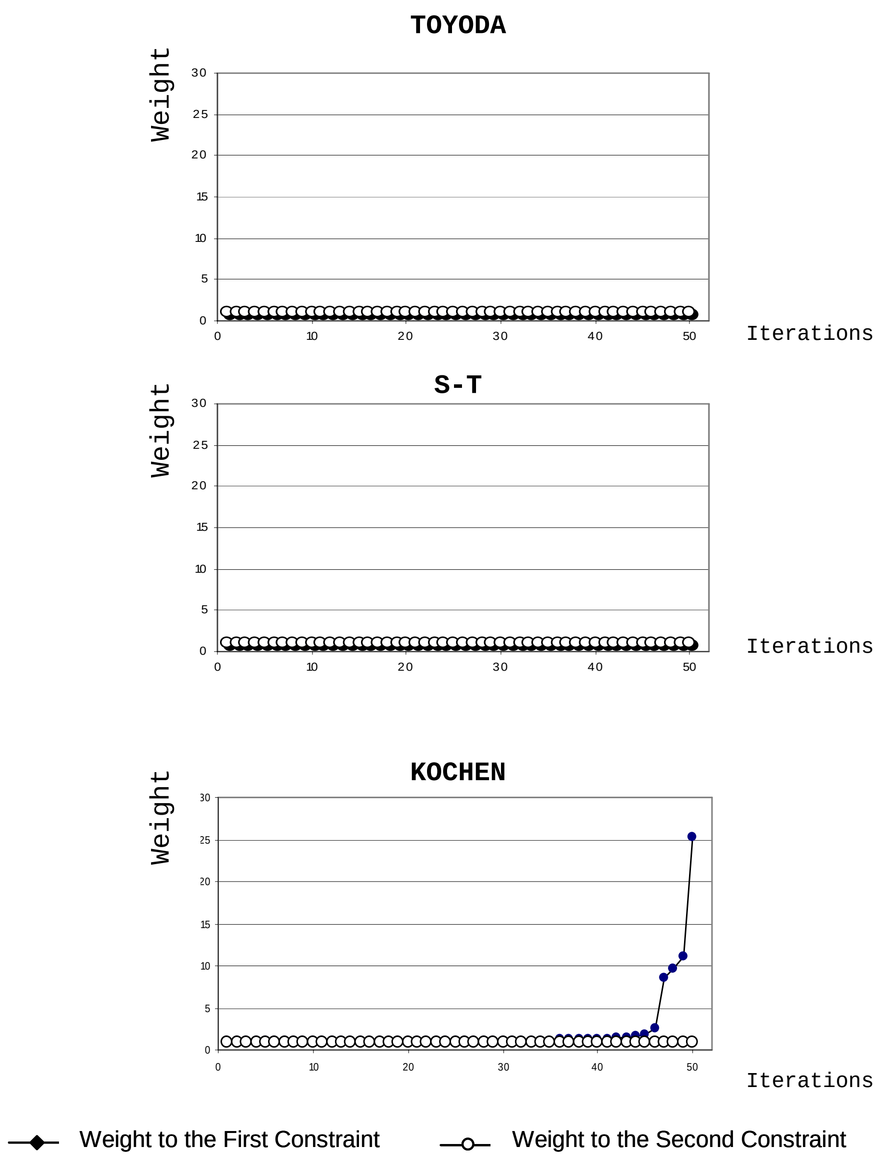


Figure 12. Weight Trend of TOYODA, S – T, KOCHEN for Correlation Structure (2, 2, 2)

In Figure 12, all weight trend lines are scaled to a minimum value of 1. Figure 12 indicates that TOYODA and S – T give the same weight in all solution iterations, while KOCHEN increases weight during the later iterations. Each resource is evenly used in the final solution (Only 0.3 % difference in resource usage by each constraint).

Table 15 shows those nine correlation structures with large differences in percentage resource remaining between the two constraints in the optimal solution, along with the best performing heuristic and the difference in resource usage when the best performing heuristic is used. Note the large (≥ 2) difference in coded correlation levels between ρ_{CA^1} and ρ_{CA^2}. That constraint with the higher positive correlation relationship with the objective function is a dominant constraint. The FOX heuristic and KOCHEN heuristic perform well under these conditions.

Table 15. Correlation Structures with More Than 10% Difference in Resource Usage by Each Constraint and Best Performing Heuristic Results

Corr Structure	%Difference in Resource Usage	Best Performing Heuristic	Corr Structure	%Difference in Resource Usage	Best Performing Heuristic	Corr Structure	%Difference in Resource Usage	Best Performing Heuristic
2,0,0	10.2 %	FOX	1,-2,-1	11.9 %	KOCHEN	-2,2,-2	32.3 %	FOX
2,-1,-1	11.7 %	FOX	0,2,0	12.5 %	FOX	-2,1,-1	11.6 %	KOCHEN
2,-2,-2	28.5 %	FOX	-1,2,-1	18.6 %	FOX	-2,0,0	10.8 %	FOX, KOCHEN

Thus, when a 2KP possesses equivalent slackness levels, problem correlation structure can create a dominant constraint. If neither constraint dominates by correlation structure, a delayed weighting scheme like KOCHEN's is suitable. If one constraint is dominant, due to correlation structure, absolute weighting on the dominant constraint as in FOX becomes a more suitable approach.

3.5.4 Influence Combinations between 2KP Slackness and Correlation

Table 16 shows the resource usage by the optimal solution in each constraint by mixed slackness settings over all correlation structures.

The data in Table 16 clearly suggests constraint slackness is a dominant consideration in how 2KP resources are used as the higher resource usage is associated with the tight constraint. Two lone exceptions are the extreme cases of ρ = (2, -2, -2) and **S** = (2, 1) and ρ = (-2, 2, -2) and **S** = (1, 2). In these two cases, the combination of perfect negative correlation between the tight constraint and the objective function means resources in the loose constraint are used at the same rate. Note, however, that even in these cases, the tight constraint is still filled to a fairly high level. The resources are evenly used in each constraint and KOCHEN is the best performing heuristic.

Table 16. Identifying Dominant Constraints by Mixed Slackness and All Correlation Structures in Optimal Solutions

Corr Structure	Slackness (1, 2)			Slackness (2,1)		
	Resource Usage in 1st Const	Resource Usage in 2nd Const	Best Performing Heuristic	Resource Usage in 1st Const	Resource Usage in 2nd Const	Best Performing Heuristic
2,2,2	1.000	0.455	FOX	0.465	1.000	N/A
2,1,1	1.000	0.501	FOX	0.735	1.000	TOYODA, S – T, KOCHEN
2,0,0	1.000	0.377	FOX	0.890	0.999	TOYODA, S – T, KOCHEN
2,-1,-1	1.000	0.508	FOX	0.992	0.999	TOYODA, S – T, KOCHEN
2,-2,-2	1.000	0.666	FOX, KOCHEN	1.000	0.983	KOCHEN
1,2,1	1.000	0.731	S – T	0.494	1.000	FOX
1,1,2	1.000	0.430	N/A	0.424	1.000	N/A
1,1,1	1.000	0.601	S – T, KOCHEN	0.609	1.000	S – T, FOX, KOCHEN
1,1,0	1.000	0.767	S – T	0.710	1.000	S – T, KOCHEN
1,0,1	1.000	0.448	S – T, FOX KOCHEN	0.666	1.000	S – T, KOCHEN
1,0,0	1.000	0.600	S – T, KOCHEN	0.776	1.000	S – T
1,0,-1	1.000	0.749	S – T	0.876	1.000	S – T
1,-1,0	1.000	0.496	S – T, KOCHEN	0.789	1.000	TOYODA, S – T, KOCHEN
1,-1,-1	1.000	0.634	S – T, KOCHEN	0.909	1.000	S – T
1,-1,-2	1.000	0.789	S – T, KOCHEN	0.968	1.000	S – T, KOCHEN
1,-2,-1	1.000	0.483	S – T, KOCHEN	0.927	0.983	S – T, L – M M1
0,2,0	1.000	0.873	TOYODA, S – T, KOCHEN	0.396	1.000	FOX
0,1,1	0.998	0.660	S – T, KOCHEN	0.428	1.000	FOX, KOCHEN
0,1,0	1.000	0.751	S – T, KOCHEN	0.588	1.000	S – T, KOCHEN
0,1,-1	0.999	0.882	S – T	0.731	1.000	S – T, KOCHEN
0,0,2	0.999	0.455	N/A	0.412	1.000	N/A
0,0,1	1.000	0.570	S – T, KOCHEN	0.539	1.000	S – T, FOX, KOCHEN
0,0,0	0.999	0.655	S – T	0.655	1.000	S – T
0,0,-1	1.000	0.790	S – T	0.760	1.000	S – T
0,0,-2	1.000	0.860	S – T	0.907	1.000	S – T
0,-1,1	1.000	0.412	N/A	0.644	0.999	TOYODA, S – T, FOX
0,-1,0	0.999	0.543	S – T, KOCHEN	0.681	0.999	S – T, KOCHEN
0,-1,-1	1.000	0.670	S – T, KOCHEN	0.782	1.000	S – T, KOCHEN
0,-2,0	1.000	0.447	FOX	0.731	0.988	S – T, L – M M1, KOCHEN
-1,2,-1	0.999	0.988	N/A	0.465	1.000	FOX
-1,1,0	1.000	0.802	S – T, KOCHEN	0.446	1.000	S – T, FOX, KOCHEN
-1,1,-1	1.000	0.885	S - T	0.626	1.000	S – T, KOCHEN
-1,1,-2	0.999	0.973	N/A	0.706	1.000	S – T, KOCHEN
-1,0,1	1.000	0.654	N/A	0.395	1.000	N/A
-1,0,0	1.000	0.712	S – T	0.554	1.000	S – T, KOCHEN
-1,0,-1	1.000	0.814	S – T, KOCHEN	0.611	1.000	S – T, KOCHEN
-1,-1,2	0.998	0.427	N/A	0.471	1.000	N/A
-1,-1,1	0.999	0.538	N/A	0.523	0.999	S – T, FOX, KOCHEN
-1,-1,0	0.998	0.632	S – T, KOCHEN	0.623	1.000	N/A
-1,-2,1	0.998	0.441	N/A	0.585	0.985	N/A
-2,2,-2	0.974	1.000	KOCHEN	0.468	1.000	FOX
-2,1,-1	0.987	0.911	S – T, L – M M1	0.435	1.000	KOCHEN
-2,0,0	0.971	0.753	L - M M1	0.397	1.000	S – T, FOX, KOCHEN
-2,-1,1	0.978	0.640	N/A	0.382	1.000	S – T, FOX, KOCHEN
-2,-2,2	0.979	0.467	N/A	0.430	0.979	N/A

3.6 Empirical Analyses Based on 5KP

3.6.1 Problem Generation for 5KP

An objective of the research is to examine heuristic performance as a function of constraint slackness and problem correlation structure. For 5KPs, each problem has 5 constraints and *n* items.

The correlation terms ρ_{CA^i} , and $\rho_{A^iA^j}$ (i , j = 1, …, 5 and $i < j$) are used in the experimental setting. The terms ρ_{CA^i} represent the correlation between objective function coefficients and *i*th constraint coefficients. The term $\rho_{A^iA^j}$ represents the correlation value between the *i*th and *j*th constraint coefficients. For constraint slackness, slackness ratios again equal 0.3 or 0.7.

However, the design of experiment for 5KP requires limitations not necessary in the 2KP case. A full factorial design for 5KPs involves 5 constraint slackness factors with 2 levels, 15 correlation structure factors with 5 levels, yielding a full factorial design of 976,562,500,000 combinations, without any replications. This is impractical.

As shown in the previous section, the solution quality of legacy heuristics is affected by problem characteristics such as various constraint slackness and correlation structures. For 5KP experiments, an objective is to confirm whether or not heuristic performance on larger problems is similar to heuristic performance on the 2KP. For example, will the primal effective gradient type methods (*e.g.*, KOCHEN) perform best when all five constraint slackness settings are equal and will the dual effective gradient type method (*e.g.*, S – T) have the best performance when the constraint slackness settings are mixed?

Figure 13 isolates greedy heuristic performance on the 2KP mixed slackness setting problems. Since three types of greedy heuristics have been discussed so far: (1) primal heuristics that considers all constraints when doing variable selection (TOYODA, KOCHEN), (2) dual heuristic (S – T), and (3) primal heuristics that consider only one constraint when doing variable selection (L – M M1, FOX), three representative heuristics, TOYODA, S – T, and L – M M1, are shown in Figure 13.

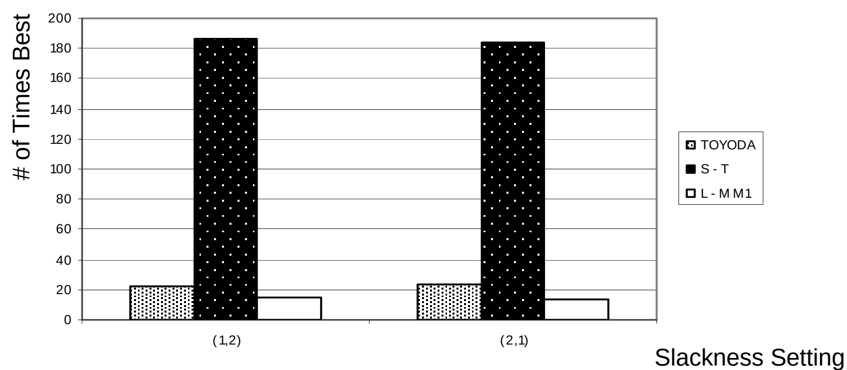


Figure 13. Elected Heuristic Performance under Slackness (1, 2) and (2, 1)

Since the marginal distribution used in the 2KP (and subsequently in the 5KP) do not vary greatly, it is not significant which constraints are defined as tight and which are defined as loose. This simplifies the experimental design and focuses the analysis effort. A similar argument holds when setting ρ_{CA^i}.

Experimental Design for 5KP Problem Generation

For 5KP, two constraint levels are considered: tight and loose. A full factorial design involves $2^5 = 32$ combinations, However, 6 combinations of constraint slackness considered are shown in Table 17; as before, 1 represents tight (0.3) and 2 represents loose (0.7) slackness setting.

Table 17. Combinations for Constraint Slackness Settings for 5KP

Constraint	Comb 1	Comb 2	Comb 3	Comb 4	Comb 5	Comb 6
A1	1	1	1	1	1	2
A2	1	1	1	1	2	2
A3	1	1	1	2	2	2
A4	1	1	2	2	2	2
A5	1	2	2	2	2	2

(1 = tight, 2 = loose)

For the 5KP, again consider five levels of correlations in each range, although not at extreme levels, with those values being

$$\rho_{CA^i} \in \{-0.9, -0.5, 0, 0.5, 0.9\}.$$

The study considers 126 cases of correlation structure between the objective function coefficients and the coefficients of the 5 constraints. These combinations of correlation represent how many constraints have various correlation with the objective function and are coded as {-2, -1, 0, 1, 2} and shown in Table B.1 of Appendix B.

As seen in Table B.1, each column represents the different combinations of correlation between objective function and constraint coefficients. Note, there is no attempt to control levels of $\rho_{A^i A^j}$. However, a correlation matrix must be positive semidefinite (Rousseeuw and Molenberghs, 1994), which restricts the range of $\rho_{A^i A^j}$. The following is the definition of a positive semidefintie matrix.

Definition: The matrix $\mathbf{A} \in \mathbf{R}^{n \times n}$ is a *positive semidefinite matrix* if

$$\forall \mathbf{x} \in \mathbf{R}^n \Rightarrow \mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0 \tag{36}$$

for all nonzero vectors $\mathbf{x} \in \mathbf{R}^n$.

Definition: The matrix $\mathbf{A} \in \mathbf{R}^{n \times n}$ is a *positive definite matrix* if

$$\mathbf{x}^T \mathbf{A} \mathbf{x} > 0 \tag{37}$$

for all nonzero vectors $\mathbf{x} \in \mathbf{R}^n$.

Theorem 2. If $\mathbf{A} \in \mathbf{R}^{n \times n}$ is a positive definite matrix and the column vectors of the matrix $\mathbf{X} \in \mathbf{R}^{n \times k}$ are linearly independent, then the matrix

$$\mathbf{B} = \mathbf{X}^T \mathbf{A} \mathbf{X} \in \mathbf{R}^{k \times k} \tag{38}$$

is also positive definite.

Proof: If for the nonzero vector $\mathbf{z} \in \mathbf{R}^k$ the relation

$$0 \geq \mathbf{z}^T \mathbf{B} \mathbf{z} \,,$$

holds, then

$$0 \geq \mathbf{z}^T \mathbf{B} \mathbf{z} = \mathbf{z}^T \mathbf{X}^T \mathbf{A} \mathbf{X} \mathbf{z} = (\mathbf{X}\mathbf{z})^T \mathbf{A} (\mathbf{X}\mathbf{z})$$

and from the positive definiteness of the matrix **A,** it follows that $\mathbf{X}\mathbf{z} = 0$, since the column vectors of the matrix **X** are linearly independent. From $\mathbf{X}\mathbf{z} = 0$, it follows that $\mathbf{z} = 0$. Hence from conditions $\mathbf{z} \in \mathbf{R}^k$ and $\mathbf{z} \neq 0$ it follows that $\mathbf{z}^T \mathbf{B} \mathbf{z} > 0$. *i.e.*, the matrix **B** is positive definite. ❑

(Tammeraid *et al.*, 2004)

Theorem 3. If the matrix $\mathbf{A} \in \mathbf{R}^{n \times n}$ is positive semidefinite, then all the submatrices of **A** obtained by deleting the same number of rows and columns of the matrix **A** are positive semidefinite.

Proof: Let $N = \{a_1, a_2, \ldots, a_k\}, \;\; k \le n$ denote the set of natural numbers satisfying the condition

$$1 \le a_1 < \;...< a_k \le n$$

then, **X** is a matrix derived from the identity matrix by taking the column-vectors with indices $a_1, \;...,\; a_k$:

$$\mathbf{X} = \underset{\quad a_1 \qquad\quad \cdots \qquad\quad a_k}{\begin{bmatrix} \vdots & 0 & \cdots & 0 \\ 1 & 0 & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & 1 & \ddots & 1 \\ 0 & 0 & \cdots & \vdots \end{bmatrix}} \in \mathbf{R}^{n \times k}$$

Since the column-vectors of the matrix **X** are linearly independent, the matrix $\mathbf{X}^{\mathbf{T}}\mathbf{A}\mathbf{X}$ is semipositive definite by Theorem 2. Therefore, all the submatrices of the matrix **A** obtained by deleting the same number of rows and columns of the matrix **A** are positive semidefinite. ❑

All submatrices obtained by deleting an equal number of rows and columns of a correlation matrix must be positive semidefinite. If we consider three variables *X*, *Y*, and *Z*, the correlation matrix has the form:

$$\Re = \begin{bmatrix} 1 & \rho_{XY} & \rho_{XZ} \\ \rho_{XY} & 1 & \rho_{YZ} \\ \rho_{XZ} & \rho_{YZ} & 1 \end{bmatrix} \tag{39}$$

A matrix $\mathfrak{R}$ is a correlation matrix if and only if it is positive semidefinite (PSD), meaning that $\mathbf{v}'\Re\mathbf{v} \geq 0$ for any column vector **v** (Rousseeuw and Molenberghs, 1994). Let *X* be the objective function coefficient vector, *Y* be the first constraint coefficient vector, and *Z* be the second constraint coefficient vector. When ρ_{XY} and ρ_{XZ} are specified, ρ_{YZ} is restricted by the positive semidefinite property, where the determinant of $\mathfrak{R}$ must be nonnegative:

$$\det(\Re) = 1 + 2\rho_{XY}\rho_{XZ}\rho_{YZ} - \rho_{XY}^2 - \rho_{XZ}^2 - \rho_{YZ}^2 \geq 0\,. \tag{40}$$

When ρ_{XY} and ρ_{XZ} are specified, values of ρ_{YZ} that ensure Equation (40) is satisfied:

$$\rho_{XY}\rho_{XZ} - \sqrt{\left(1-\rho_{XY}^2\right)\left(1-\rho_{XZ}^2\right)} \leq \rho_{YZ} \leq \rho_{XY}\rho_{XZ} + \sqrt{\left(1-\rho_{XY}^2\right)\left(1-\rho_{XZ}^2\right)} \tag{41}$$

Theorem 4. If a value of ρ_{YZ} is the midpoint of Equation (41), *i.e.*, $\rho_{YZ} = \rho_{XY}\rho_{XZ}$, the matrix $\mathfrak{R}$ is always a positive semidefinite matrix.

Proof: When the matrix $\mathfrak{R}$ is positive semidefinite, the determinant of $\mathfrak{R}$ must be as follows:

$$\det(\Re) = 1 + 2\rho_{XY}\rho_{XZ}\rho_{YZ} - \rho_{XY}^2 - \rho_{XZ}^2 - \rho_{YZ}^2 \geq 0$$

Substitute $$\rho_{YZ} = \rho_{XY}\rho_{XZ}$$

$$1 + 2\rho_{XY}^2\rho_{XZ}^2 - \rho_{XY}^2 - \rho_{XZ}^2 - \rho_{XY}^2\rho_{XZ}^2 \geq 0$$

$$1 + \rho_{XY}^2\rho_{XZ}^2 - \rho_{XY}^2 - \rho_{XZ}^2 \geq 0$$

$$\left(1 - \rho_{XY}^2\right)\left(1 - \rho_{XZ}^2\right) \geq 0$$

Since ρ_{XY} and ρ_{XZ} are correlation coefficients, $-1 \leq \rho_{XY} \leq 1$ and $-1 \leq \rho_{XZ} \leq 1$.

Therefore, if $\rho_{YZ} = \rho_{XY}\rho_{XZ}$, matrix $\mathfrak{R}$ is a positive semidefinite matrix. ❑

The result of Theorem 4 provides a convenient way to set any $\rho_{A^iA^j}$ once ρ_{CA^i} and ρ_{CA^j} are specified from the set $[-0.9, \ -0.5, \ 0, \ 0.5, \ 0.9]$. Use

$$\rho_{A^iA^j} = \rho_{CA^i} \cdot \rho_{CA^j}. \qquad (42)$$

Given a correlation structure between objective function coefficients and constraint coefficients, the following "Procedure CorrGeneration" is used to create the 5KP problem set.

Procedure CorrGeneration

1. Specify the desired correlation matrix $\mathfrak{R}$ in terms of ρ_{CA^i}, $i = 1, \ldots, 5$.
2. Calculate each $\rho_{A^iA^j}$ using $\rho_{A^iA^j} = \rho_{CA^i}\rho_{CA^j}$.
3. Fully specify the correlation matrix $\mathfrak{R}$ (6×6 matrix):

$$\Re = \begin{bmatrix} 1 & \rho_{CA^1} & \rho_{CA^2} & \rho_{CA^3} & \rho_{CA^4} & \rho_{CA^5} \\ \rho_{CA^1} & 1 & \rho_{A^1A^2} & \rho_{A^1A^3} & \rho_{A^1A^4} & \rho_{A^1A^5} \\ \rho_{CA^2} & \rho_{A^1A^2} & 1 & \rho_{A^2A^3} & \rho_{A^2A^4} & \rho_{A^2A^5} \\ \rho_{CA^3} & \rho_{A^1A^3} & \rho_{A^2A^3} & 1 & \rho_{A^3A^4} & \rho_{A^3A^5} \\ \rho_{CA^4} & \rho_{A^1A^4} & \rho_{A^2A^4} & \rho_{A^3A^4} & 1 & \rho_{A^4A^5} \\ \rho_{CA^5} & \rho_{A^1A^5} & \rho_{A^2A^5} & \rho_{A^3A^5} & \rho_{A^4A^5} & 1 \end{bmatrix}$$

4. Substitute into generation routine (Procedure Iman and Cononver Approach) to get 5KP problem set.

Iman and Conover (1982) suggest how to induce a Spearman correlation structure given by the correlation matrix $\Re$ among a set of random variables. The following Iman and Conover approach ensures that the final Spearman correlation matrix **M** of the input vectors is close to the desired correlation matrix $\Re$, while preserving the marginal distribution of the input vectors.

Procedure Iman and Conover approach (1982) adapted for 5KP problems with correlation structure

1. Perform Cholesky factorization of the desired correlation matrix $\Re$ (6×6 matrix) to get a lower matrix **P**: $\Re = \mathbf{P} \circ \mathbf{P}^{\mathrm{T}}$.
2. Generate 6 random number vectors with which to randomize the Van der Waerden Scores (VW scores).
3. Create the VW score matrix, **H,** using VW Scores: VW Scores = $\Phi^{-1}\left(\frac{i}{N+1}\right)$ where i = order, N = 100, Φ^{-1} = the inverse CDF of the standard normal distribution.
4. Create the sample correlation matrix **T** based on **H**.

5. Cholesky factorization of **T** to get a lower matrix $\mathbf{Q}$ by $\mathbf{T} = \mathbf{Q} \circ \mathbf{Q}^{\mathrm{T}}$.

6. Create the transformation matrix **S**: $\mathbf{S} = \mathbf{P} \circ \mathbf{Q}^{-1}$.

7. Create the new matrix $\mathbf{H}^*$: $\mathbf{H}^* = \mathbf{H} \circ \mathbf{S}^{\mathrm{T}}$.

8. Extract each column vector, compute the ranks, and place these ranks into the rank matrix **M** (6×100 matrix).

9. Generate 6 random number vectors (100 dimensions of each vector) following the marginal distribution of input variables (the C, A^1, A^2, A^3, A^4, and A^5 vectors)

10. Sort random numbers by descending order.

11. Rearrange random numbers in Step 10 following the rank (specified order) in the **M** matrix.

For the 5KP, the number of constraints is 5 and the number of variables is 100. The objective function coefficients, c_j, are integer numbers uniformly distributed from 1 to 100. The coefficients of the first constraint, a_{1j}, are integer numbers uniformly distributed from 1 to 55, the coefficients of the second constraint, a_{2j}, are integer numbers uniformly distributed from 1 to 35, the coefficients of the third constraint, a_{3j}, are integer numbers uniformly distributed from 1 to 30, the coefficients of the fourth constraint, a_{4j}, are integer numbers uniformly distributed from 1 to 45, and the coefficients of the fifth constraint, a_{5j}, are integer numbers uniformly distributed from 1 to 25. Each uniform distribution makes sure each constraint has a different value for its greatest coefficient.

For the correlation structure, the five levels of correlation for each correlation term ρ_{CA^i} are coded as {-2, -1, 0, 1, 2} and slackness levels of 0.3 or 0.7 are coded as 1 or 2, respectively.

The experimental design for 5KP problem generation employs the 6 constraint slackness settings as shown in Table 17 and varies the correlation structures among objective function and constraint coefficients using 126 separate correlation structures as shown in Table B.1 of Appendix B. Each design point is replicated 5 times to yield a total of 3780 (6 × 126 × 5) test problems that effectively varied all problem characteristics across the entire range of values. Thus, the various correlation structures are blended across each static slackness setting.

3.6.2 Heuristic Performance Based on 5KP Constraint Slackness

The overall performance of the legacy greedy heuristics on the newly generated 5KP problems is summarized in Table 18. Tables C.1 and C.2 in Appendix C provide the detailed statistical test results that are summarized in Table 18.

Table 18. Number of Times Best by Each Heuristic under 5KP Constraint Slackness

Heuristics	**(1,1,1,1,1)**	**(1,1,1,1,2)**	**(1,1,1,2,2)**	**(1,1,2,2,2)**	**(1,2,2,2,2)**	**(2,2,2,2,2)**
TOYODA	8	15	22	22	16	5
S – T	1	4	20	49	198	1
L – M M1	105	92	96	109	74	60
FOX	61	31	26	15	12	114
KOCHEN	437	452	408	326	168	412
Statistical Tests						
Chi-Square Test	Reject $\mathbf{H}_0^C$	Reject $\mathbf{H}_0^C$	Reject $\mathbf{H}_0^C$	Reject $\mathbf{H}_0^C$	Reject $\mathbf{H}_0^C$	Reject $\mathbf{H}_0^C$
Best by Sign Test	**KOCHEN**	**KOCHEN**	**KOCHEN**	**KOCHEN**	**S – T and KOCHEN**	**KOCHEN**

($\mathbf{H}_0^C$: Heuristic performances do not differ. Reject Region: α =0.1)

The data reflect the various combinations of constraint slackness ratios and summarize how many times each heuristic is the outright best performer. Chi-square test and sign test are used to examine best heuristics. There are clearly differences in performance.

2KP empirical analysis on varied constraint slackness settings suggests that the better greedy heuristics give more consideration to a dominant constraint. This allows heuristics to select more items for inclusion without violating the dominant constraint, and thus improve the objective function value attained. The analysis of 5KP constraint slackness shows that a dominant constraint produces the same effect on heuristic performance as in the 2KP and this is stated in the following hypothesis:

Hypothesis of 5KP Slackness

Varied levels of constraint slackness in an MKP will dictate any dominant constraint (When slackness levels vary, then the tighter constraints are the dominant constraints).

A dominant constraint is defined as that constraint whose solution feasibility drives the solution for the entire problem. When there is a single dominant constraint, such as with (1, 2, 2, 2, 2), the S – T appears slightly favored while KOCHEN is best for the other constraint slackness settings examined.

The results in Table 18 find that KOCHEN is the best performer under similar constraint slackness settings, which are **S** = (1, 1, 1, 1, 1) and **S** = (2, 2, 2, 2, 2) in the 5KP case. When there is one dominant constraint such as **S** = (1, 2, 2, 2, 2), S – T appears to be the slightly better heuristic. However, three important questions arise. First, why is KOCHEN the best overall performer except under specific constraint slackness setting **S** = (1, 2, 2, 2, 2)? Second, why are S – T and KOCHEN similar for constraint slackness setting **S** = (1, 2, 2, 2, 2) but not any of the other five settings? Third, why does the L – M M1 heuristic performance seem to improve in the 5KP test set as compared to the 2KP test set results?

The dynamic penalty cost functions (weighting schemes) used in each heuristic are analyzed to help answer the questions posed. Thus, a slackness setting **S** = (1, 1, 2, 2, 2) is picked to be examined to determine why KOCHEN is best on all slackness settings except **S** = (1, 2, 2, 2, 2). Figure 14 plots the average variation in weight vector per iteration for slackness setting **S** = (1, 1, 2, 2, 2) problems and shows three different weighting schemes: TOYODA's early weighting scheme, S – T's constant weighting scheme, and KOCHEN's delayed weighting scheme. Each line in the graphs represents the weight trend line for each of the five problem constraints. For ease of reference, all weights are scaled to a minimum value of 1. As with the 2KP analysis, KOCHEN's delayed weighting scheme is a better weighting scheme as item selection is more difficult in the later iterations. Since fewer items can fit into tight constraints, compared to loose constraints, proper item choice based on tight constraints is important. Table 19 summarizes average resource usage by the heuristics' solution and the optimal solution for setting **S** = (1, 1, 2, 2, 2) under correlation structure ρ = (0, 0, 0, 0, 0). Since the focus is the effect of constraint slackness on heuristic performance, only 5KP problems with independence among problem coefficients are considered.

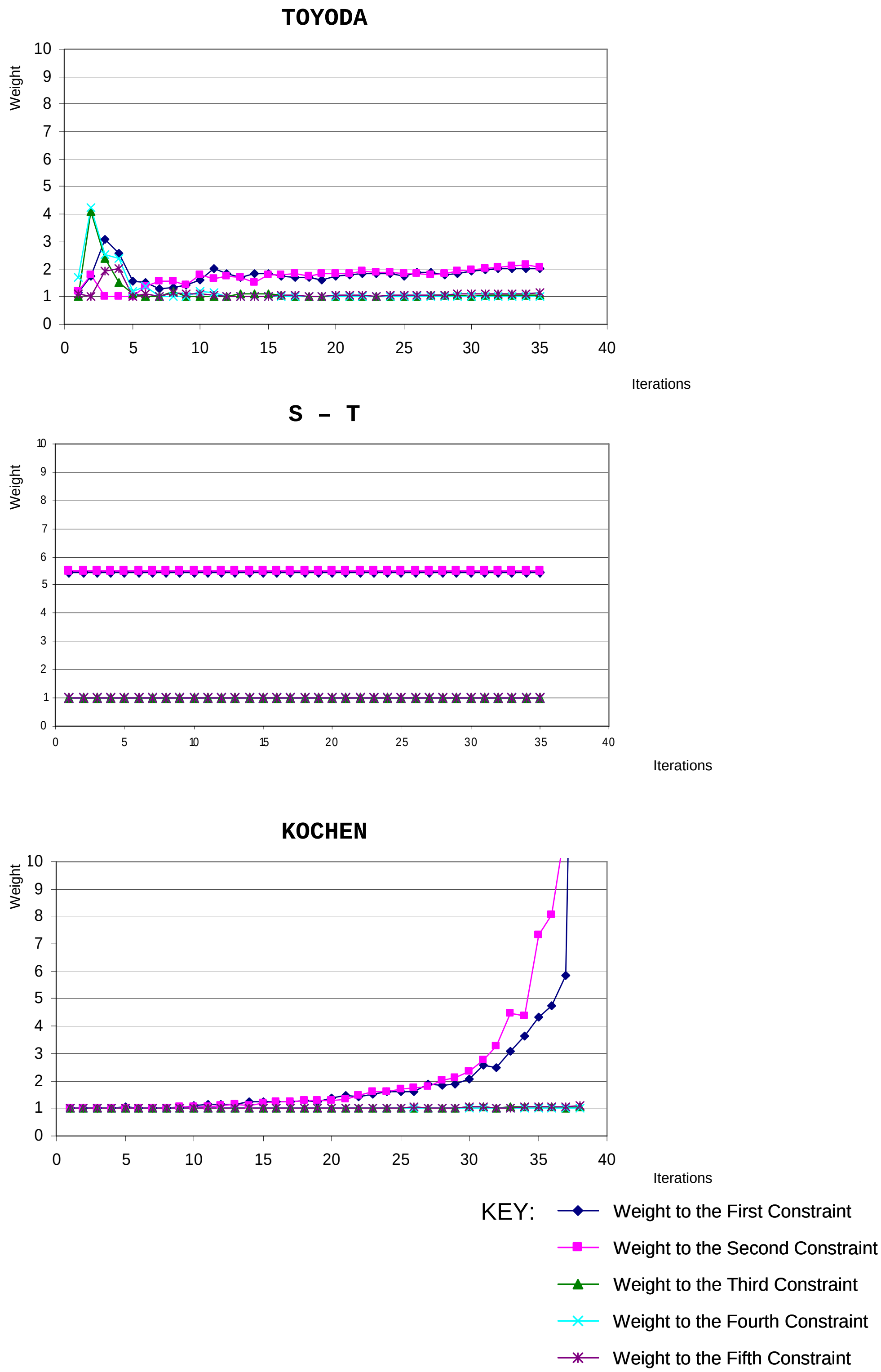


Figure 14. Performance of Weight Trend of Legacy Heuristics for Setting (1,1,2,2,2)

Table 19 shows that the dominant constraints are determined by constraint slackness settings. (Table C.3 in Appendix C provides resource usage for all slackness settings under zero correlation)

Table 19. Resource Usage in Slackness Setting (1, 1, 2, 2, 2) under Zero Correlation.

Heuristics	# of Vars Selected	Resource Usage 1st Const	Resource Usage 2nd Const	Resource Usage 3rd Const	Resource Usage 4th Const	Resource Usage 5th Const
TOYODA	38.4	0.976	0.993	0.516	0.530	0.522
S – T	39.4	0.977	0.995	0.548	0.555	0.556
KOCHEN	40.2	0.987	0.991	0.556	0.574	0.557
L – M M1	36.4	0.988	0.993	0.510	0.543	0.521
FOX	40.2	0.994	0.991	0.566	0.587	0.551
Optimal	39.4	0.998	0.997	0.564	0.565	0.556

(Ratio of Resource Usage)

The Table 19 results show FOX has the best performance from a resource usage perspective; FOX consumes the resources most effectively in the tight constraints. Thus, FOX can select more variables while maintaining feasibility than other heuristics; however, this conclusion is not supported by the results in Table 18. This points to the criticality of choosing the correct item; KOCHEN picks as many items as FOX but it picks those items more effectively. Thus, the delayed weighting scheme of KOCHEN, seen in Figure 14, is the most effective.

To answer the second question (Why are S – T and KOCHEN similar for constraint slackness setting (1, 2, 2, 2, 2) but not any of the other five settings?), Figure 15 plots the average weight values of TOYODA, S – T, and KOCHEN for constraint slackness setting (1, 2, 2, 2, 2).

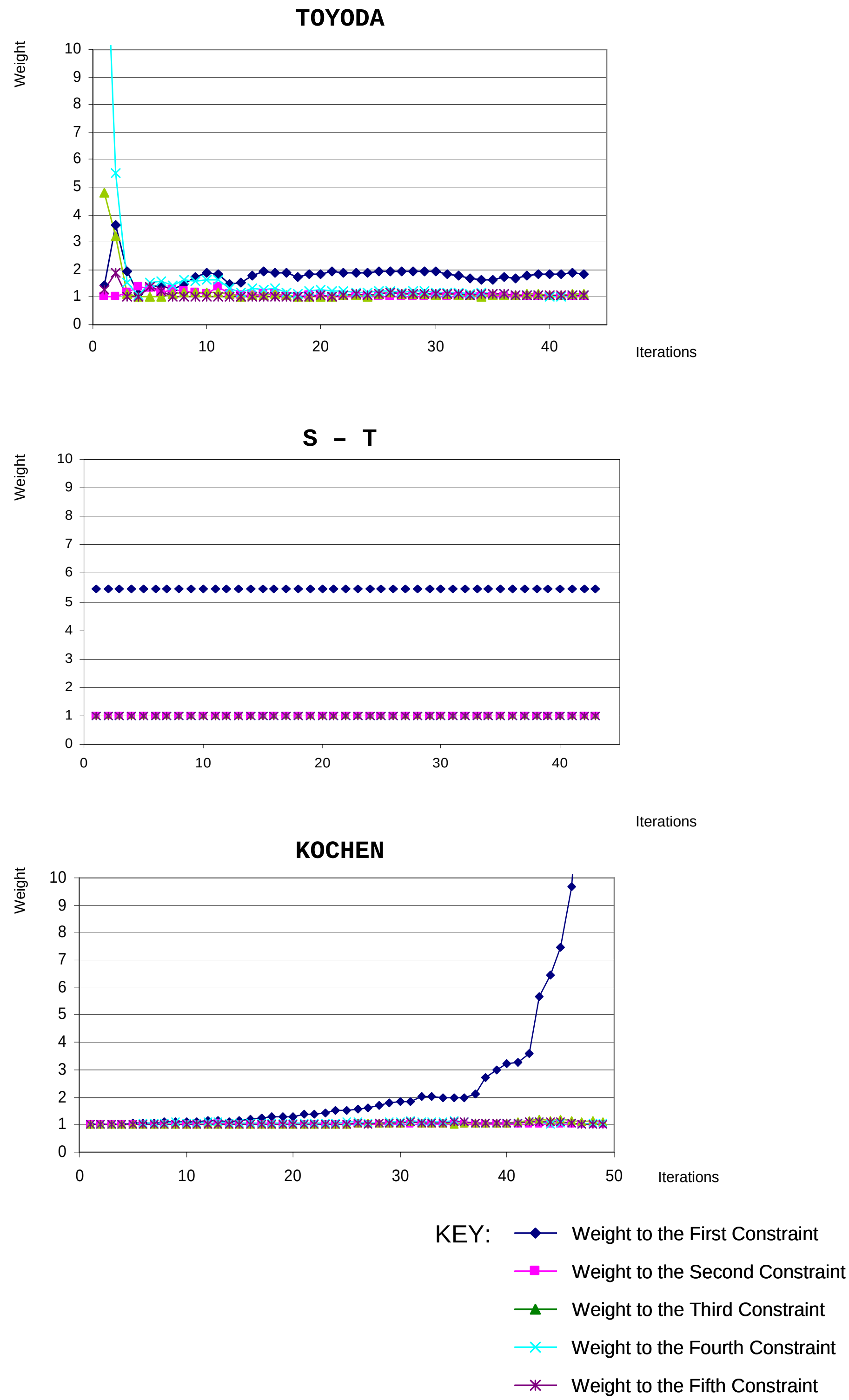


Figure 15. Performance of Weight Trend of Legacy Heuristics for (1,2,2,2,2)

In this type of problem, the first constraint is the dominant constraint and should not only have the least resources remaining during the final iterations, but also during almost all iterations in the solution process. To accommodate a dominant constraint, an effective heuristic needs to focus variable selection with respect to that specific dominant constraint. As depicted in Figure 15, S – T always gives the dominant constraint more weight. However, KOCHEN rapidly increases the weight applied to the dominant constraint during the latter stages of the process. This significantly increases focus on the dominant constraint later in the process and allows KOCHEN to achieve results similar to S – T.

Table 20. Resource Usage in Slackness Setting (1, 2, 2, 2, 2) under Zero Correlation.

Heuristics	# of Vars Selected	Resource Usage 1st Const	Resource Usage 2nd Const	Resource Usage 3rd Const	Resource Usage 4th Const	Resource Usage 5th Const
TOYODA	40.8	0.998	0.564	0.564	0.542	0.55
S – T	44.2	0.998	0.62	0.625	0.603	0.613
KOCHEN	44.6	0.997	0.628	0.629	0.612	0.629
L – M M1	42.2	0.996	0.599	0.613	0.593	0.615
FOX	45.6	0.992	0.643	0.654	0.638	0.647
Optimal	45.6	1	0.655	0.659	0.624	0.658

(Ratio of Resource Usage)

Table 20 summarizes resource usage of each heuristic for problem **S** = (1, 2, 2, 2, 2) and ρ = (0, 0, 0, 0, 0). Each heuristic nearly fills the tight constraint. The FOX heuristic is again a best choice for overall resource usage but is not a good choice for obtaining a best solution. The FOX heuristic does not always pick the proper variables to set to 1 (add to the knapsack). The constant weighting scheme of S – T and the late weighting scheme of KOCHEN in the solution process ensure selection of the best items for the knapsacks.

To answer the third question (Why does the L – M M1 heuristic performance seem to improve with a 5KP test set as compared to a 2KP test set?), comparing L – M M1 with FOX, L – M M1 considers the future potential resource usage in each constraint for an item selection. This characteristic tries to balance resource usage on all constraints which is similar to TOYODA's weighting scheme. Since there are many combinations between constraint slackness and correlation structures in the 5KP, considering all constraints for an item selection in the equal slackness settings yields a better solution than considering only the most limiting constraint, as in FOX, at each iteration. Thus, as the number of constraints increases, L – M M1 performs better than FOX. The computational tests in Chapter VI increase the number of constraints up to 25. This computational test shows that L – M M1 outperforms FOX as the number of constraints increases.

3.6.3 Heuristic Performance Based on 5KP Correlation Structures

Based on the analysis of 2KP heuristics, if all constraints have similar slackness levels, positive correlation between objective function and constraint coefficients can dictate heuristic performance. When constraint slackness levels are mixed, then positive correlation is not as dominant a consideration. In all 2KP cases, the KOCHEN delayed weighting scheme showed the most robust performance except for the extreme correlation structures $\rho = (2, -2, *)$ and $\rho = (-2, 2, *)$ where $*$ represents any value of

$\rho_{A^1A^2}$. For these extreme cases, the FOX absolute weighting scheme is more suitable. The following analysis tests this conjecture.

Hypothesis of 5KP Correlation Structure

When an MKP possesses equivalent slackness levels, problem correlation structure can dictate a dominant constraint.

Table 21 shows the best performer by correlation structures under equal slackness in the 5KP case, Table 22 shows the average relative errors by each heuristic and Table 23 shows the resource usage by KOCHEN under **S** = (1, 1, 1, 1, 1). The resource usage by KOCHEN under **S** = (1, 1, 1, 1, 2) is provided in Table 26. In addition, Tables C.5 through C.8 in Appendix C provide resource usage by KOCHEN under other slackness settings and Table C.4 presents the detailed statistical tests to discriminate a best heuristic.

Table 21. Best Performer Counts by Correlation Structure under Equal Slackness in 5KP

Correlation	TOYODA	S – T	L – M M1	FOX	KOCHEN	χ^2 Test Reject $\mathbf{H}_0^c$	Best by Sign Test	Total Probs
-2,-2,-2,-2,-2	0	0	2	2	2	N	N/A	10
-2,-2,-2,-2,-1	0	0	3	1	4	Y	L – M M1, KOCHEN	10
-2,-2,-2,-2,0	0	0	2	6	0	Y	FOX	10
-2,-2,-2,-2,1	0	0	2	6	1	Y	L – M M1, FOX	10
-2,-2,-2,-2,2	0	0	0	9	1	Y	FOX	10
-2,-2,-2,-1,-1	0	0	4	1	3	Y	L – M M1, KOCHEN	10
-2,-2,-2,-1,0	0	0	6	2	2	Y	L-M M1	10
-2,-2,-2,-1,1	0	0	5	3	2	Y	L-M M1, FOX, KOCHEN	10
-2,-2,-2,-1,2	0	0	0	10	0	Y	FOX	10
-2,-2,-2,0,0	1	0	0	0	8	Y	KOCHEN	10
-2,-2,-2,0,1	0	0	0	0	10	Y	KOCHEN	10
-2,-2,-2,0,2	0	0	0	8	2	Y	FOX	10
-2,-2,-2,1,1	0	0	0	0	10	Y	KOCHEN	10
-2,-2,-2,1,2	0	0	2	4	4	Y	FOX, KOCHEN	10
-2,-2,-2,2,2	0	0	0	0	10	Y	KOCHEN	10
-2,-2,-1,-1,-1	0	0	2	1	4	Y	L – M M1, KOCHEN	10
-2,-2,-1,-1,0	0	0	4	1	4	Y	L – M M1, KOCHEN	10
-2,-2,-1,-1,1	0	0	6	4	0	Y	L-M M1, FOX	10
-2,-2,-1,-1,2	0	0	0	8	0	Y	FOX	10
-2,-2,-1,0,0	0	0	0	0	9	Y	KOCHEN	10
-2,-2,-1,0,1	0	0	4	0	6	Y	L – M M1, KOCHEN	10
-2,-2,-1,0,2	0	0	0	8	2	Y	FOX	10
-2,-2,-1,1,1	0	0	0	0	9	Y	KOCHEN	10
-2,-2,-1,1,2	0	0	2	4	3	N	N/A	10
-2,-2,-1,2,2	0	0	0	0	10	Y	KOCHEN	10
-2,-2,0,0,0	1	0	1	0	7	Y	KOCHEN	10
-2,-2,0,0,1	0	0	2	0	8	Y	KOCHEN	10
-2,-2,0,0,2	0	0	1	5	4	Y	FOX, KOCHEN	10
-2,-2,0,1,1	0	0	0	0	10	Y	KOCHEN	10
-2,-2,0,1,2	0	0	3	3	4	N	N/A	10
-2,-2,0,2,2	0	0	0	0	10	Y	KOCHEN	10
-2,-2,1,1,1	0	0	0	0	10	Y	KOCHEN	10
-2,-2,1,1,2	0	0	3	0	7	Y	KOCHEN	10
-2,-2,1,2,2	0	0	0	0	10	Y	KOCHEN	10
-2,-2,2,2,2	0	0	0	0	10	Y	KOCHEN	10
-2,-1,-1,-1,-1	0	0	5	0	5	Y	L – M M1, KOCHEN	10
-2,-1,-1,-1,0	0	0	9	0	1	Y	L-M M1	10
-2,-1,-1,-1,1	0	0	2	3	5	Y	KOCHEN	10
-2,-1,-1,-1,2	0	0	2	7	1	Y	FOX	10
-2,-1,-1,0,0	0	0	1	0	8	Y	KOCHEN	10
-2,-1,-1,0,1	0	0	3	0	7	Y	KOCHEN	10
-2,-1,-1,0,2	0	0	1	9	0	Y	FOX	10
-2,-1,-1,1,1	0	0	0	0	10	Y	KOCHEN	10
-2,-1,-1,1,2	0	0	1	4	5	Y	FOX, KOCHEN	10
-2,-1,-1,2,2	0	0	0	1	9	Y	KOCHEN	10
-2,-1,0,0,0	0	0	1	0	9	Y	KOCHEN	10
-2,-1,0,0,1	0	0	3	1	6	Y	KOCHEN	10
-2,-1,0,0,2	0	0	3	6	1	Y	FOX	10
-2,-1,0,1,1	0	0	0	0	10	Y	KOCHEN	10
-2,-1,0,1,2	0	0	0	5	5	Y	FOX, KOCHEN	10
-2,-1,0,2,2	0	0	0	0	10	Y	KOCHEN	10
-2,-1,1,1,1	0	0	0	0	10	Y	KOCHEN	10
-2,-1,1,1,2	0	0	2	2	6	Y	KOCHEN	10
-2,-1,1,2,2	0	0	1	0	9	Y	KOCHEN	10
-2,-1,2,2,2	0	0	1	0	9	Y	KOCHEN	10
-2,0,0,0,0	0	1	0	0	6	Y	KOCHEN	10
-2,0,0,0,1	0	0	3	0	6	Y	L – M M1, KOCHEN	10
-2,0,0,0,2	0	0	2	4	4	Y	FOX, KOCHEN	10
-2,0,0,1,1	0	0	0	0	10	Y	KOCHEN	10
-2,0,0,1,2	0	0	2	2	6	Y	KOCHEN	10
-2,0,0,2,2	0	0	0	0	10	Y	KOCHEN	10
-2,0,1,1,1	0	0	0	0	10	Y	KOCHEN	10

Correlation	TOYODA	S – T	L – M M1	FOX	KOCHEN	χ^2 Test Reject $\mathbf{H}_0^c$	Best by Sign Test	Total Probs
-2,0,1,1,2	0	0	2	2	6	Y	KOCHEN	10
-2,0,1,2,2	0	0	1	0	9	Y	KOCHEN	10
-2,0,2,2,2	0	0	0	0	10	Y	KOCHEN	10
-2,1,1,1,1	3	0	0	0	6	Y	TOYODA, KOCHEN	10
-2,1,1,1,2	0	0	1	0	9	Y	KOCHEN	10
-2,1,1,2,2	0	0	0	0	10	Y	KOCHEN	10
-2,1,2,2,2	0	0	0	0	10	Y	KOCHEN	10
-2,2,2,2,2	2	0	0	0	8	Y	KOCHEN	10
-1,-1,-1,-1,-1	0	0	1	0	2	**N**	N/A	10
-1,-1,-1,-1,0	0	0	7	0	1	Y	L-M M1	10
-1,-1,-1,-1,1	0	0	4	3	3	**N**	N/A	10
-1,-1,-1,-1,2	0	0	1	7	2	Y	FOX	10
-1,-1,-1,0,0	0	0	1	0	9	Y	KOCHEN	10
-1,-1,-1,0,1	0	0	1	0	8	Y	KOCHEN	10
-1,-1,-1,0,2	0	0	1	7	2	Y	FOX	10
-1,-1,-1,1,1	0	0	0	0	10	Y	KOCHEN	10
-1,-1,-1,1,2	0	0	2	2	6	Y	KOCHEN	10
-1,-1,-1,2,2	0	0	0	0	10	Y	KOCHEN	10
-1,-1,0,0,0	0	0	2	0	6	Y	KOCHEN	10
-1,-1,0,0,1	0	0	2	2	6	Y	KOCHEN	10
-1,-1,0,0,2	0	0	2	3	5	Y	KOCHEN	10
-1,-1,0,1,1	0	0	0	0	10	Y	KOCHEN	10
-1,-1,0,1,2	0	0	0	5	5	Y	FOX, KOCHEN	10
-1,-1,0,2,2	0	0	0	1	9	Y	KOCHEN	10
-1,-1,1,1,1	1	0	0	0	9	Y	KOCHEN	10
-1,-1,1,1,2	0	0	3	1	6	Y	KOCHEN	10
-1,-1,1,2,2	0	0	0	0	10	Y	KOCHEN	10
-1,-1,2,2,2	0	0	0	0	9	Y	KOCHEN	10
-1,0,0,0,0	1	0	0	0	7	Y	KOCHEN	10
-1,0,0,0,1	0	0	4	0	6	Y	L – M M1, KOCHEN	10
-1,0,0,0,2	0	0	4	3	2	**N**	N/A	10
-1,0,0,1,1	0	0	0	0	10	Y	KOCHEN	10
-1,0,0,1,2	0	0	3	1	6	Y	KOCHEN	10
-1,0,0,2,2	0	0	0	0	10	Y	KOCHEN	10
-1,0,1,1,1	0	0	0	0	10	Y	KOCHEN	10
-1,0,1,1,2	0	0	1	0	9	Y	KOCHEN	10
-1,0,1,2,2	0	0	2	0	8	Y	KOCHEN	10
-1,0,2,2,2	0	0	0	0	10	Y	KOCHEN	10
-1,1,1,1,1	2	0	0	0	8	Y	KOCHEN	10
-1,1,1,1,2	0	0	5	1	4	Y	L-M M1, KOCHEN	10
-1,1,1,2,2	0	0	1	0	9	Y	KOCHEN	10
-1,1,2,2,2	0	0	0	0	10	Y	KOCHEN	10
-1,2,2,2,2	1	0	0	0	9	Y	KOCHEN	10
0,0,0,0,0	0	0	1	0	2	**N**	N/A	10
0,0,0,0,1	0	0	3	0	7	Y	KOCHEN	10
0,0,0,0,2	0	0	2	3	5	Y	KOCHEN	10
0,0,0,1,1	0	0	1	0	9	Y	KOCHEN	10
0,0,0,1,2	0	0	1	2	7	Y	KOCHEN	10
0,0,0,2,2	0	0	0	0	10	Y	KOCHEN	10
0,0,1,1,1	1	0	0	0	9	Y	KOCHEN	10
0,0,1,1,2	0	0	2	2	6	Y	KOCHEN	10
0,0,1,2,2	0	0	2	0	7	Y	KOCHEN	10
0,0,2,2,2	0	0	0	0	10	Y	KOCHEN	10
0,1,1,1,1	0	0	0	0	10	Y	KOCHEN	10
0,1,1,1,2	0	0	1	0	9	Y	KOCHEN	10
0,1,1,2,2	0	0	0	0	10	Y	KOCHEN	10
0,1,2,2,2	0	0	0	0	10	Y	KOCHEN	10
0,2,2,2,2	0	0	0	0	10	Y	KOCHEN	10
1,1,1,1,1	0	0	0	0	7	Y	KOCHEN	10
1,1,1,1,2	0	0	4	0	6	Y	L –M M1, KOCHEN	10
1,1,1,2,2	0	0	1	0	9	Y	KOCHEN	10
1,1,2,2,2	0	0	0	0	10	Y	KOCHEN	10
1,2,2,2,2	0	0	0	0	10	Y	KOCHEN	10
2,2,2,2,2	0	1	0	0	8	Y	KOCHEN	10

(Reject Region: α =0.1)

Table 22. Relative Errors by Correlation Structure under Equal Slackness in 5KP

Correlation	TOYODA	S – T	L – M M1	FOX	KOCHEN
-2,-2,-2,-2,-2	0.546	0.546	0.183	0.191	0.292
-2,-2,-2,-2,-1	1.595	2.083	0.396	0.828	0.360
-2,-2,-2,-2,0	4.003	5.438	1.148	0.752	1.173
-2,-2,-2,-2,1	5.818	8.577	1.776	1.838	1.913
-2,-2,-2,-2,2	7.990	12.468	5.202	2.233	4.048
-2,-2,-2,-1,-1	1.220	2.050	0.915	1.679	0.706
-2,-2,-2,-1,0	3.338	5.518	0.971	1.916	1.238
-2,-2,-2,-1,1	6.264	9.441	2.166	2.837	2.480
-2,-2,-2,-1,2	7.254	10.131	5.256	1.142	4.105
-2,-2,-2,0,0	2.121	2.410	1.767	3.609	0.563
-2,-2,-2,0,1	5.827	8.455	2.095	5.203	1.264
-2,-2,-2,0,2	8.406	10.807	4.813	2.351	3.709
-2,-2,-2,1,1	3.251	6.058	3.314	5.654	0.697
-2,-2,-2,1,2	9.192	10.794	4.487	4.074	3.775
-2,-2,-2,2,2	4.923	9.180	4.911	4.494	2.263
-2,-2,-1,-1,-1	1.029	1.011	0.602	2.644	0.384
-2,-2,-1,-1,0	3.334	5.156	0.609	3.602	0.639
-2,-2,-1,-1,1	6.177	9.629	1.432	2.578	1.949
-2,-2,-1,-1,2	7.876	10.275	4.810	2.246	3.936
-2,-2,-1,0,0	2.609	3.658	1.700	5.023	0.731
-2,-2,-1,0,1	5.276	7.661	2.221	4.700	1.683
-2,-2,-1,0,2	6.870	9.272	4.235	2.784	3.692
-2,-2,-1,1,1	3.556	6.347	3.275	7.281	1.383
-2,-2,-1,1,2	7.058	10.306	4.938	4.915	3.479
-2,-2,-1,2,2	5.347	8.371	5.746	4.978	2.133
-2,-2,0,0,0	1.015	1.584	1.993	7.227	0.506
-2,-2,0,0,1	4.935	7.336	2.428	6.866	1.394
-2,-2,0,0,2	10.028	13.976	5.488	4.860	4.457
-2,-2,0,1,1	2.921	4.958	3.020	7.512	0.935
-2,-2,0,1,2	7.983	9.801	4.626	4.848	4.112
-2,-2,0,2,2	5.452	8.563	5.333	5.665	2.239
-2,-2,1,1,1	2.162	3.503	4.385	8.454	0.582
-2,-2,1,1,2	6.106	7.746	3.140	5.029	2.748
-2,-2,1,2,2	4.915	7.778	4.950	6.347	1.834
-2,-2,2,2,2	3.257	7.218	4.604	6.779	0.841
-2,-1,-1,-1,-1	0.993	1.437	0.936	3.474	0.537
-2,-1,-1,-1,0	2.904	3.181	0.652	4.310	1.163
-2,-1,-1,-1,1	7.095	10.362	2.057	4.436	1.905
-2,-1,-1,-1,2	8.730	11.631	5.210	3.943	4.087
-2,-1,-1,0,0	2.288	2.964	1.822	4.625	0.922
-2,-1,-1,0,1	5.606	7.812	2.289	4.951	1.591
-2,-1,-1,0,2	6.944	8.756	4.258	2.545	3.719
-2,-1,-1,1,1	2.528	4.632	3.399	6.331	1.115
-2,-1,-1,1,2	8.593	11.022	4.378	4.383	3.572
-2,-1,-1,2,2	5.581	9.349	4.881	5.638	2.238
-2,-1,0,0,0	1.987	1.982	1.453	7.800	0.559
-2,-1,0,0,1	6.453	8.588	2.490	6.696	1.677
-2,-1,0,0,2	8.329	10.119	3.836	3.127	4.179
-2,-1,0,1,1	3.755	5.075	3.381	8.025	1.172
-2,-1,0,1,2	7.667	9.236	4.489	3.091	3.776
-2,-1,0,2,2	3.948	5.881	4.223	4.463	1.945
-2,-1,1,1,1	2.728	3.561	3.934	9.308	0.998
-2,-1,1,1,2	6.941	8.737	4.400	5.744	3.383
-2,-1,1,2,2	0.546	0.546	0.183	0.191	0.292
-2,-1,2,2,2	1.595	2.083	0.396	0.828	0.360
-2,0,0,0,0	4.003	5.438	1.148	0.752	1.173
-2,0,0,0,1	5.818	8.577	1.776	1.838	1.913
-2,0,0,0,2	7.990	12.468	5.202	2.233	4.048
-2,0,0,1,1	1.220	2.050	0.915	1.679	0.706
-2,0,0,1,2	3.338	5.518	0.971	1.916	1.238
-2,0,0,2,2	6.264	9.441	2.166	2.837	2.480
-2,0,1,1,1	7.254	10.131	5.256	1.142	4.105
-2,0,1,1,2	2.121	2.410	1.767	3.609	0.563

Correlation	TOYODA	S – T	L – M M1	FOX	KOCHEN
-2,0,1,2,2	5.630	8.112	5.465	5.971	1.990
-2,0,2,2,2	2.720	5.797	3.253	5.392	1.009
-2,1,1,1,1	1.130	1.472	1.596	7.554	0.263
-2,1,1,1,2	4.042	6.821	1.819	6.304	1.578
-2,1,1,2,2	8.684	10.667	5.284	5.497	4.287
-2,1,2,2,2	3.025	6.024	2.828	9.275	1.300
-2,2,2,2,2	8.584	10.206	4.906	5.737	3.082
-1,-1,-1,-1,-1	4.867	6.794	4.789	5.355	1.862
-1,-1,-1,-1,0	2.292	3.177	2.906	9.920	0.656
-1,-1,-1,-1,1	6.604	7.821	4.017	5.729	2.928
-1,-1,-1,-1,2	5.744	7.790	4.600	6.584	2.022
-1,-1,-1,0,0	3.451	5.997	5.037	5.733	1.096
-1,-1,-1,0,1	1.391	1.657	3.832	9.373	0.631
-1,-1,-1,0,2	7.238	9.226	4.984	7.665	3.206
-1,-1,-1,1,1	4.739	7.200	4.749	6.251	1.799
-1,-1,-1,1,2	4.020	5.909	5.215	5.947	1.388
-1,-1,-1,2,2	2.262	4.695	4.412	6.528	0.899
-1,-1,0,0,0	0.840	1.137	0.839	3.628	0.371
-1,-1,0,0,1	3.444	3.939	0.597	3.928	0.961
-1,-1,0,0,2	6.819	10.215	1.994	4.645	2.496
-1,-1,0,1,1	7.845	10.532	4.182	3.020	4.007
-1,-1,0,1,2	3.208	4.695	2.608	6.079	0.713
-1,-1,0,2,2	5.868	8.397	2.469	4.922	0.999
-1,-1,1,1,1	9.820	12.300	4.834	4.399	3.684
-1,-1,1,1,2	3.719	5.462	3.144	7.325	1.661
-1,-1,1,2,2	8.753	11.462	5.461	4.877	3.899
-1,-1,2,2,2	5.233	7.447	4.716	5.091	1.892
-1,0,0,0,0	1.954	2.027	1.702	7.101	0.821
-1,0,0,0,1	4.959	6.916	2.080	6.698	1.556
-1,0,0,0,2	8.003	9.959	4.706	4.318	3.634
-1,0,0,1,1	4.389	6.223	2.713	7.264	1.263
-1,0,0,1,2	7.853	9.441	4.526	4.261	3.248
-1,0,0,2,2	5.640	7.779	4.762	4.940	2.290
-1,0,1,1,1	2.860	4.326	3.956	10.066	0.809
-1,0,1,1,2	7.267	10.400	4.829	6.263	3.312
-1,0,1,2,2	4.003	6.205	3.820	5.487	1.611
-1,0,2,2,2	3.093	5.284	3.931	5.540	1.167
-1,1,1,1,1	1.584	1.940	1.637	7.421	0.717
-1,1,1,1,2	6.136	7.231	1.807	7.905	1.542
-1,1,1,2,2	8.299	9.643	3.633	4.766	3.835
-1,1,2,2,2	4.453	5.632	3.404	7.817	1.645
-1,2,2,2,2	8.323	10.817	4.723	5.405	3.805
0,0,0,0,0	5.086	6.509	4.643	5.668	1.971
0,0,0,0,1	2.250	3.443	3.178	8.643	0.725
0,0,0,0,2	7.640	8.684	4.257	5.830	3.227
0,0,0,1,1	5.485	8.428	5.154	6.065	2.070
0,0,0,1,2	3.421	6.537	3.967	6.654	1.502
0,0,0,2,2	1.392	2.787	3.239	11.981	0.659
0,0,1,1,1	6.973	8.089	3.139	6.020	3.185
0,0,1,1,2	4.603	6.063	4.820	6.714	1.949
0,0,1,2,2	4.685	5.674	4.686	6.349	1.356
0,0,2,2,2	1.703	2.886	2.904	7.322	0.679
0,1,1,1,1	0.703	0.875	1.646	8.601	0.640
0,1,1,1,2	6.241	7.374	1.863	8.806	1.490
0,1,1,2,2	8.130	9.148	4.293	4.325	3.735
0,1,2,2,2	3.621	5.731	3.638	8.318	1.548
0,2,2,2,2	8.498	10.019	5.202	6.116	4.093
1,1,1,1,1	5.583	7.260	6.044	5.843	3.012
1,1,1,1,2	2.098	2.605	3.236	8.802	0.853
1,1,1,2,2	7.403	8.498	4.480	7.472	3.279
1,1,2,2,2	5.731	7.368	3.822	7.014	2.263
1,2,2,2,2	3.548	5.386	3.824	8.248	1.271
2,2,2,2,2	1.503	2.114	3.855	10.426	0.688

Unit: Percent

Table 23. Resource Usage by KOCHEN under Slackness Setting (1, 1, 1, 1, 1)

Correlation	1st Const.	2nd Const.	3rd Const.	4th Const.	5th Const.
-2,-2,-2,-2,-2	0.956	0.930	0.948	0.924	0.957
-2,-2,-2,-2,-1	0.859	0.920	0.887	0.898	0.998
-2,-2,-2,-2,0	0.794	0.797	0.774	0.792	0.999
-2,-2,-2,-2,1	0.660	0.668	0.639	0.692	1.000
-2,-2,-2,-2,2	0.477	0.462	0.491	0.463	1.000
-2,-2,-2,-1,-1	0.793	0.798	0.812	0.970	0.991
-2,-2,-2,-1,0	0.714	0.699	0.707	0.941	0.996
-2,-2,-2,-1,1	0.626	0.652	0.654	0.812	1.000
-2,-2,-2,-1,2	0.416	0.439	0.420	0.499	1.000
-2,-2,-2,0,0	0.664	0.629	0.656	0.987	0.991
-2,-2,-2,0,1	0.624	0.625	0.615	0.965	0.999
-2,-2,-2,0,2	0.473	0.461	0.489	0.670	1.000
-2,-2,-2,1,1	0.644	0.676	0.665	0.995	1.000
-2,-2,-2,1,2	0.556	0.588	0.594	0.854	1.000
-2,-2,-2,2,2	0.551	0.505	0.509	0.993	0.999
-2,-2,-1,-1,-1	0.772	0.732	0.969	0.981	0.980
-2,-2,-1,-1,0	0.709	0.743	0.954	0.947	0.998
-2,-2,-1,-1,1	0.639	0.646	0.823	0.805	1.000
-2,-2,-1,-1,2	0.466	0.479	0.605	0.564	1.000
-2,-2,-1,0,0	0.649	0.698	0.860	0.988	0.994
-2,-2,-1,0,1	0.604	0.603	0.775	0.918	1.000
-2,-2,-1,0,2	0.460	0.489	0.604	0.694	1.000
-2,-2,-1,1,1	0.612	0.599	0.766	0.986	0.998
-2,-2,-1,1,2	0.542	0.543	0.581	0.896	1.000
-2,-2,-1,2,2	0.558	0.563	0.634	0.997	1.000
-2,-2,0,0,0	0.559	0.567	0.979	0.983	0.980
-2,-2,0,0,1	0.677	0.655	0.935	0.942	0.998
-2,-2,0,0,2	0.618	0.640	0.704	0.724	1.000
-2,-2,0,1,1	0.630	0.619	0.906	0.996	0.988
-2,-2,0,1,2	0.479	0.484	0.682	0.847	1.000
-2,-2,0,2,2	0.599	0.586	0.702	0.999	0.992
-2,-2,1,1,1	0.599	0.627	0.988	0.991	0.994
-2,-2,1,1,2	0.534	0.534	0.856	0.848	1.000
-2,-2,1,2,2	0.596	0.595	0.832	0.996	0.993
-2,-2,2,2,2	0.622	0.635	0.989	0.998	0.995
-2,-1,-1,-1,-1	0.753	0.958	0.979	0.939	0.979
-2,-1,-1,-1,0	0.687	0.921	0.932	0.912	0.998
-2,-1,-1,-1,1	0.672	0.837	0.856	0.885	1.000
-2,-1,-1,-1,2	0.570	0.627	0.665	0.625	1.000
-2,-1,-1,0,0	0.644	0.865	0.806	0.993	0.994
-2,-1,-1,0,1	0.674	0.861	0.826	0.972	0.998
-2,-1,-1,0,2	0.472	0.569	0.574	0.726	1.000
-2,-1,-1,1,1	0.610	0.737	0.776	1.000	0.997
-2,-1,-1,1,2	0.478	0.609	0.569	0.888	1.000
-2,-1,-1,2,2	0.621	0.620	0.664	0.993	0.999
-2,-1,0,0,0	0.624	0.848	0.976	0.993	0.993
-2,-1,0,0,1	0.613	0.738	0.919	0.948	0.999
-2,-1,0,0,2	0.482	0.581	0.674	0.660	1.000
-2,-1,0,1,1	0.691	0.803	0.920	0.996	0.994
-2,-1,0,1,2	0.432	0.559	0.680	0.837	1.000
-2,-1,0,2,2	0.539	0.597	0.699	0.998	0.998
-2,-1,1,1,1	0.559	0.751	0.990	0.976	0.991
-2,-1,1,1,2	0.534	0.663	0.884	0.843	1.000
-2,-1,1,2,2	0.659	0.716	0.877	0.987	0.999
-2,-1,2,2,2	0.551	0.624	0.995	0.991	0.991
-2,0,0,0,0	0.579	0.990	0.979	0.996	0.988
-2,0,0,0,1	0.666	0.942	0.944	0.948	0.995
-2,0,0,0,2	0.622	0.715	0.716	0.731	1.000
-2,0,0,1,1	0.602	0.918	0.877	0.997	0.990
-2,0,0,1,2	0.596	0.767	0.757	0.897	1.000
-2,0,0,2,2	0.652	0.718	0.742	0.998	0.995
-2,0,1,1,1	0.600	0.872	0.988	0.993	0.997
-2,0,1,1,2	0.549	0.759	0.843	0.848	1.000

Correlation	1st Const.	2nd Const.	3rd Const.	4th Const.	5th Const.
-2,0,1,2,2	0.567	0.770	0.809	0.994	1.000
-2,0,2,2,2	0.687	0.753	0.997	0.993	0.995
-2,1,1,1,1	0.647	0.983	0.986	0.979	0.986
-2,1,1,1,2	0.613	0.879	0.858	0.835	1.000
-2,1,1,2,2	0.703	0.895	0.847	0.996	0.999
-2,1,2,2,2	0.789	0.863	0.989	0.996	0.987
-2,2,2,2,2	0.692	0.983	0.977	0.978	0.993
-1,-1,-1,-1,-1	0.961	0.989	0.960	0.968	0.963
-1,-1,-1,-1,0	0.900	0.926	0.935	0.895	0.996
-1,-1,-1,-1,1	0.807	0.824	0.780	0.799	1.000
-1,-1,-1,-1,2	0.547	0.589	0.584	0.555	1.000
-1,-1,-1,0,0	0.887	0.891	0.924	0.999	0.998
-1,-1,-1,0,1	0.811	0.869	0.874	0.975	1.000
-1,-1,-1,0,2	0.680	0.688	0.711	0.729	1.000
-1,-1,-1,1,1	0.763	0.757	0.778	0.993	0.993
-1,-1,-1,1,2	0.638	0.673	0.643	0.865	1.000
-1,-1,-1,2,2	0.634	0.637	0.620	1.000	0.992
-1,-1,0,0,0	0.831	0.838	0.995	0.974	0.975
-1,-1,0,0,1	0.804	0.769	0.952	0.961	0.996
-1,-1,0,0,2	0.602	0.607	0.709	0.765	1.000
-1,-1,0,1,1	0.759	0.771	0.910	0.992	0.997
-1,-1,0,1,2	0.635	0.602	0.733	0.890	1.000
-1,-1,0,2,2	0.628	0.638	0.681	0.994	0.999
-1,-1,1,1,1	0.787	0.809	0.989	0.990	0.996
-1,-1,1,1,2	0.600	0.624	0.854	0.892	1.000
-1,-1,1,2,2	0.608	0.634	0.821	0.999	0.988
-1,-1,2,2,2	0.582	0.621	0.992	0.989	0.995
-1,0,0,0,0	0.861	0.982	0.970	0.985	0.967
-1,0,0,0,1	0.732	0.946	0.939	0.929	1.000
-1,0,0,0,2	0.629	0.768	0.735	0.734	1.000
-1,0,0,1,1	0.751	0.895	0.900	0.995	0.993
-1,0,0,1,2	0.641	0.767	0.749	0.885	1.000
-1,0,0,2,2	0.650	0.681	0.740	0.998	0.995
-1,0,1,1,1	0.738	0.911	0.989	0.997	0.991
-1,0,1,1,2	0.615	0.693	0.873	0.856	1.000
-1,0,1,2,2	0.745	0.831	0.885	0.998	0.994
-1,0,2,2,2	0.658	0.703	0.987	0.993	0.988
-1,1,1,1,1	0.748	0.987	0.983	0.986	0.990
-1,1,1,1,2	0.689	0.899	0.870	0.881	1.000
-1,1,1,2,2	0.627	0.865	0.817	0.996	0.995
-1,1,2,2,2	0.759	0.874	0.994	0.991	0.994
-1,2,2,2,2	0.629	0.989	0.990	0.983	0.994
0,0,0,0,0	0.962	0.967	0.956	0.982	0.974
0,0,0,0,1	0.897	0.901	0.949	0.927	0.998
0,0,0,0,2	0.703	0.712	0.736	0.729	1.000
0,0,0,1,1	0.881	0.872	0.878	0.982	0.993
0,0,0,1,2	0.750	0.730	0.761	0.902	1.000
0,0,0,2,2	0.761	0.706	0.772	0.996	0.993
0,0,1,1,1	0.874	0.886	0.988	0.985	0.989
0,0,1,1,2	0.741	0.815	0.904	0.897	1.000
0,0,1,2,2	0.773	0.701	0.846	0.995	1.000
0,0,2,2,2	0.721	0.723	0.995	0.994	0.993
0,1,1,1,1	0.882	0.977	0.994	0.994	0.973
0,1,1,1,2	0.770	0.860	0.879	0.868	1.000
0,1,1,2,2	0.749	0.842	0.861	0.991	0.998
0,1,2,2,2	0.769	0.872	0.991	0.992	0.984
0,2,2,2,2	0.783	0.992	0.997	0.990	0.993
1,1,1,1,1	0.987	0.978	0.961	0.975	0.978
1,1,1,1,2	0.884	0.919	0.882	0.890	1.000
1,1,1,2,2	0.872	0.847	0.867	0.995	0.998
1,1,2,2,2	0.860	0.902	0.996	0.996	0.988
1,2,2,2,2	0.866	0.992	0.984	0.997	0.988
2,2,2,2,2	0.991	0.988	0.990	0.982	0.993

Analysis for Hypothesis of 5KP Correlation Structure

There are five different coded correlation values for the five constraints: {-2, -1, 0, 1, 2}. Coded correlation value 2, ρ_{CA^i} = 0.9, makes the *i*th constraint the dominant constraint. A constraint with ρ_{CA^i} = 0.9 has its resources used up earlier than other constraints. Table 23 supports the statement that constraints possessing positive correlation with the objective function coefficients have a higher percentage of resources used. Table 13 showed this information for the 2KP.

Based on the analysis of 2KP correlation structures where constraints have similar correlation values (less than a difference of 1 among coded correlation value for each constraint pair), resources are used evenly in the optimal solution, and KOCHEN is the best performing heuristic. Table 24 shows those correlation structures having similar difference (less than 3% difference) in resource usage between the smallest resource remaining constraint and the second smallest resource remaining constraint in the optimal solution, along with the best performing heuristic. This means there are at least two dominant constraints for the associated correlation structure.

Table 24. Correlation Structures Having Similar Resource Usage (Less than 3% Difference) Between the Smallest and the Next Smallest Constraint and Best Performing Heuristic

Correlation Structure	Best Performing Heuristic	Correlation Structure	Best Performing Heuristic	Correlation Structure	Best Performing Heuristic	Correlation Structure	Best Performing Heuristic
-2,-2,-2,-2,-2	ALL	-2,-1,-1,-1,1	KOCHEN	-2,1,2,2,2	KOCHEN	-1,1,1,1,1	KOCHEN
-2,-2,-2,-1,-1	L – M M1, KOCHEN	-2,-1,-1,0,0	KOCHEN	-2,2,2,2,2	KOCHEN	-1,1,1,1,2	L-M M1, KOCHEN
-2,-2,-2,-1,0	L-M M1	-2,-1,-1,0,1	KOCHEN	-1,-1,-1,-1,-1	ALL	-1,1,1,2,2	KOCHEN
-2,-2,-2,-1,1	L-M M1, FOX, KOCHEN	-2,-1,-1,1,1	KOCHEN	-1,-1,-1,-1,0	L-M M1	-1,1,2,2,2	KOCHEN
-2,-2,-2,0,0	KOCHEN	-2,-1,-1,1,2	FOX, KOCHEN	-1,-1,-1,0,0	KOCHEN	-1,2,2,2,2	KOCHEN
-2,-2,-2,0,1	KOCHEN	-2,-1,-1,2,2	KOCHEN	-1,-1,-1,0,1	KOCHEN	0,0,0,0,0	ALL
-2,-2,-2,1,1	KOCHEN	-2,-1,0,0,0	KOCHEN	-1,-1,-1,1,1	KOCHEN	0,0,0,0,1	KOCHEN
-2,-2,-2,1,2	FOX, KOCHEN	-2,-1,0,0,1	KOCHEN	-1,-1,-1,1,2	KOCHEN	0,0,0,1,1	KOCHEN
-2,-2,-2,2,2	KOCHEN	-2,-1,0,1,1	KOCHEN	-1,-1,-1,2,2	KOCHEN	0,0,0,1,2	KOCHEN
-2,-2,-1,-1,-1	L – M M1, KOCHEN	-2,-1,0,1,2	FOX, KOCHEN	-1,-1,0,0,0	KOCHEN	0,0,0,2,2	KOCHEN
-2,-2,-1,-1,0	L – M M1, KOCHEN	-2,-1,0,2,2	KOCHEN	-1,-1,0,0,1	KOCHEN	0,0,1,1,1	KOCHEN
-2,-2,-1,0,0	KOCHEN	-2,-1,1,1,1	KOCHEN	-1,-1,0,1,1	KOCHEN	0,0,1,1,2	KOCHEN
-2,-2,-1,0,1	L – M M1, KOCHEN	-2,-1,1,1,2	KOCHEN	-1,-1,0,1,2	FOX, KOCHEN	0,0,1,2,2	KOCHEN
-2,-2,-1,1,1	KOCHEN	-2,-1,1,2,2	KOCHEN	-1,-1,0,2,2	KOCHEN	0,0,2,2,2	KOCHEN
-2,-2,-1,1,2	ALL	-2,-1,2,2,2	KOCHEN	-1,-1,1,1,1	KOCHEN	0,1,1,1,1	KOCHEN
-2,-2,-1,2,2	KOCHEN	-2,0,0,0,0	KOCHEN	-1,-1,1,1,2	KOCHEN	0,1,1,1,2	KOCHEN
-2,-2,0,0,0	KOCHEN	-2,0,0,0,1	L – M M1, KOCHEN	-1,-1,1,2,2	KOCHEN	0,1,1,2,2	KOCHEN
-2,-2,0,0,1	KOCHEN	-2,0,0,0,2	FOX, KOCHEN	-1,-1,2,2,2	KOCHEN	0,1,2,2,2	KOCHEN
-2,-2,0,0,2	FOX, KOCHEN	-2,0,0,1,1	KOCHEN	-1,0,0,0,0	KOCHEN	0,2,2,2,2	KOCHEN
-2,-2,0,1,1	KOCHEN	-2,0,0,1,2	KOCHEN	-1,0,0,0,1	L – M M1, KOCHEN	1,1,1,1,1	KOCHEN
-2,-2,0,1,2	ALL	-2,0,0,2,2	KOCHEN	-1,0,0,0,2	ALL	1,1,1,1,2	L –M M1, KOCHEN
-2,-2,0,2,2	KOCHEN	-2,0,1,1,1	KOCHEN	-1,0,0,1,1	KOCHEN	1,1,1,2,2	KOCHEN
-2,-2,1,1,1	KOCHEN	-2,0,1,1,2	KOCHEN	-1,0,0,1,2	KOCHEN	1,1,2,2,2	KOCHEN
-2,-2,1,1,2	KOCHEN	-2,0,1,2,2	KOCHEN	-1,0,0,2,2	KOCHEN	1,2,2,2,2	KOCHEN
-2,-2,1,2,2	KOCHEN	-2,0,2,2,2	KOCHEN	-1,0,1,1,1	KOCHEN	2,2,2,2,2	KOCHEN
-2,-2,2,2,2	KOCHEN	-2,1,1,1,1	TOYODA, KOCHEN	-1,0,1,1,2	KOCHEN		
-2,-1,-1,-1,-1	L – M M1, KOCHEN	-2,1,1,1,2	KOCHEN	-1,0,1,2,2	KOCHEN		
-2,-1,-1,-1,0	L-M M1	-2,1,1,2,2	KOCHEN	-1,0,2,2,2	KOCHEN		

Table 24 agrees with the analysis of 2KP correlation structures. If there are at least two constraints with similar objective function – constraint correlation values (within a difference of 1 in coded correlation values), the successful heuristic should balance the weight on these constraints in order to use evenly resources in early iterations, and then increase the weight on the constraint with the least resource remaining in the later iterations.   Thus, KOCHEN is expected to be the best performing heuristic. Table 24 indicates KOCHEN is the best performing heuristic for 106 of the 109 correlation structures. Therefore, if the correlation structure has at least two high correlation values, or all correlation values are the same, a delayed weighting scheme is the most appropriate.

Figure 16 shows the weighting trend of KOCHEN for different correlation structures.

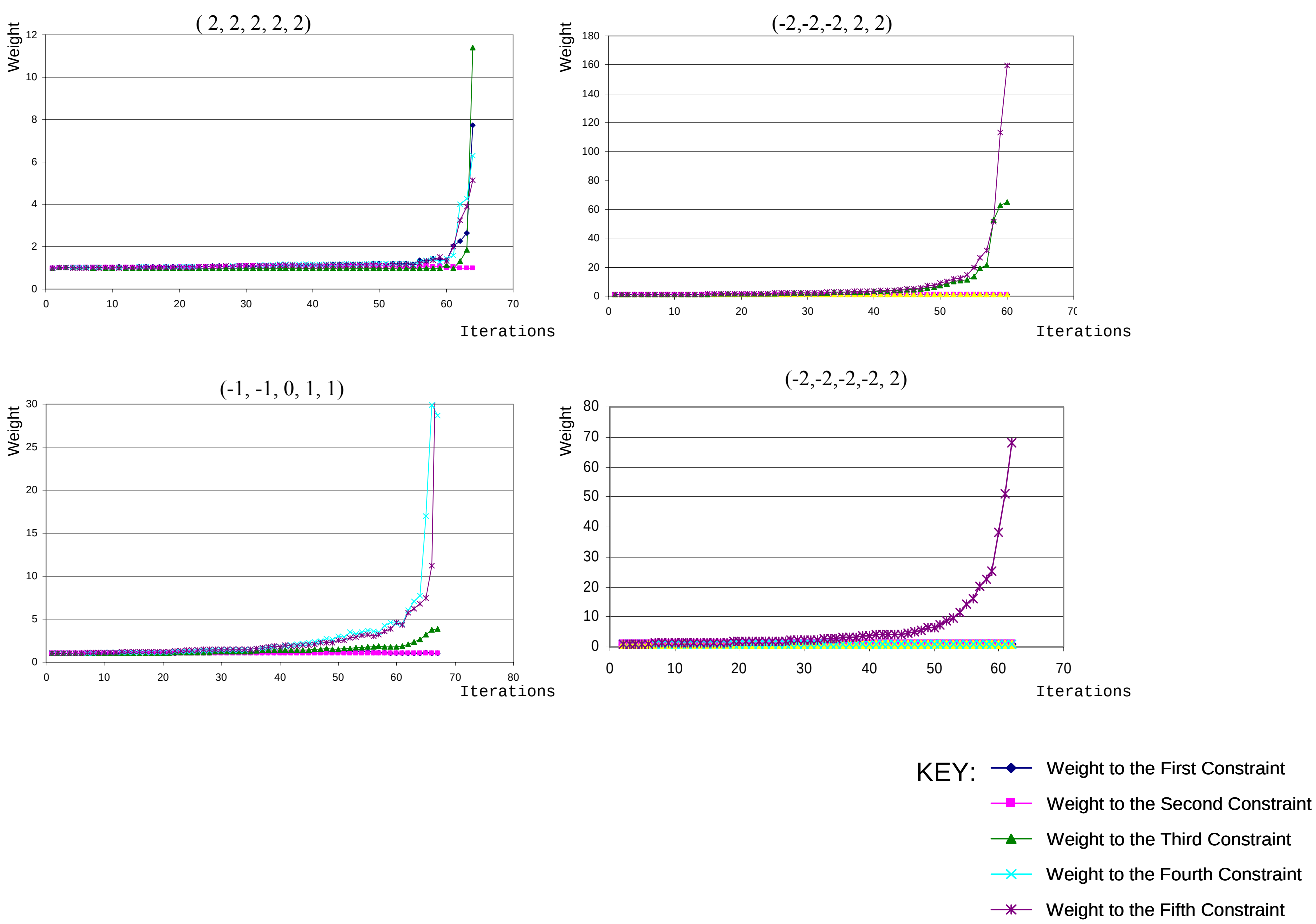


Figure 16. Weight Trend of KOCHEN according to Correlation Structure

The correlation structure (2, 2, 2, 2, 2) indicates that all constraints relate similarly to the objective function so each resource will be almost completely used. For this situation, KOCHEN gives the same weight to each constraint in almost all iterations, and then places the most weight on the constraint with the least resource remaining in the last few iterations. This characteristic allows KOCHEN to put more items into the knapsack. For correlation structures ρ = (-2,-2,-2, 2, 2) and ρ = (-1,-1, 0, 1, 1), the weights on the two dominant constraints increase rapidly in later iterations. Thus, KOCHEN selects items by comparing objective function coefficients with two dominant constraint coefficients. For correlation structure ρ = (-2,-2,-2,-2, 2), KOCHEN places more weight on the fifth constraint from the middle number of iterations until termination.

For the extreme case ρ = (-2, -2, -2, -2, 2), the fifth constraint is the dominant constraint and should have the least resource remaining in the final iterations. Table 25 shows those correlation structures having larger differences (more than 10% difference) in resource usage between the smallest resource remaining and the second smallest resource remaining, along with the best performing heuristic. In both instances, the FOX heuristic performs best.

Table 25. Correlation Structures with More Than 10% Difference in Resource Usage by the Least Constraint and the Second Least Constraint and Best Performing Heuristic

Correlation Structure	Best Performing Heuristic	Correlation Structure	Best Performing Heuristic
-2,-2,-2,-2,2	FOX	-2,-2,-2,-1,2	FOX

Thus, the Hypothesis concerning 5KP correlation structure is confirmed, and the analysis for this hypothesis agrees with the analysis of 2KP correlation structures. This means that if there are some higher correlation values, KOCHEN is the best performing heuristic. However, if there is one dominant constraint created by the correlation structure, FOX's absolute weighting is more suitable.

3.6.4 Influence Combinations between 5KP Slackness and Correlation

Table 26 supports the conjecture that constraint slackness is the dominant consideration in how resources are used for the 5KP. The extreme combinations, which are perfect positive correlation between loose constraints and objective function and negative correlation between tight constraint and objective function, are exceptions to the conjecture of slackness domination (for example $\boldsymbol{\rho}$ = (-2, -2, -2, -2, 2) and **S** = (1, 1, 1, 1, 2)). The combination of perfect positive correlation between the loose constraint and the objective function allows the loose constraint to be filled at the same rate as the tight constraint. Note, the correlation structure dictates a dominant constraint among tight constraints. For example, the combination of positive correlation between the tight constraint and the objective function makes the tight constraint a dominant constraint. These results support the hypothesis of 5KP correlation structure: Problem correlation structure can dictate a dominant constraint under equal slackness levels.

Table 26. Resource Usage by KOCHEN under Slackness Setting (1, 1, 1, 1, 2)

Correlation	1st Const.	2nd Const.	3rd Const.	4th Const.	5th Const.
-2,-2,-2,-2,-2	0.916	0.938	0.927	0.954	0.425
-2,-2,-2,-2,-1	0.915	0.922	0.972	0.943	0.563
-2,-2,-2,-2,0	0.944	0.971	0.938	0.928	0.717
-2,-2,-2,-2,1	0.951	0.960	0.927	0.897	0.855
-2,-2,-2,-2,2	0.911	0.917	0.948	0.957	0.990
-2,-2,-2,-1,-1	0.845	0.846	0.879	0.991	0.515
-2,-2,-2,-1,0	0.852	0.889	0.941	0.994	0.657
-2,-2,-2,-1,1	0.864	0.887	0.858	0.996	0.806
-2,-2,-2,-1,2	0.901	0.881	0.892	0.998	0.954
-2,-2,-2,0,0	0.831	0.771	0.846	0.997	0.568
-2,-2,-2,0,1	0.770	0.788	0.764	0.998	0.708
-2,-2,-2,0,2	0.846	0.858	0.843	0.998	0.828
-2,-2,-2,1,1	0.764	0.760	0.791	1.000	0.610
-2,-2,-2,1,2	0.723	0.730	0.749	0.999	0.698
-2,-2,-2,2,2	0.657	0.671	0.682	1.000	0.527
-2,-2,-1,-1,-1	0.816	0.848	0.993	0.994	0.487
-2,-2,-1,-1,0	0.831	0.811	0.993	0.990	0.633
-2,-2,-1,-1,1	0.837	0.807	0.983	0.990	0.800
-2,-2,-1,-1,2	0.797	0.799	0.976	0.994	0.895
-2,-2,-1,0,0	0.776	0.787	0.957	0.998	0.574
-2,-2,-1,0,1	0.738	0.749	0.920	0.998	0.714
-2,-2,-1,0,2	0.769	0.751	0.974	0.997	0.822
-2,-2,-1,1,1	0.731	0.689	0.889	0.999	0.595
-2,-2,-1,1,2	0.730	0.744	0.851	1.000	0.683
-2,-2,-1,2,2	0.607	0.598	0.632	1.000	0.505
-2,-2,0,0,0	0.713	0.716	0.995	0.996	0.543
-2,-2,0,0,1	0.710	0.738	0.992	0.985	0.672
-2,-2,0,0,2	0.726	0.737	0.991	0.988	0.767
-2,-2,0,1,1	0.749	0.691	0.985	0.999	0.582
-2,-2,0,1,2	0.751	0.720	0.978	0.996	0.669
-2,-2,0,2,2	0.563	0.566	0.715	1.000	0.487
-2,-2,1,1,1	0.735	0.733	0.994	0.994	0.565
-2,-2,1,1,2	0.713	0.724	0.993	0.992	0.645
-2,-2,1,2,2	0.546	0.567	0.840	1.000	0.499
-2,-2,2,2,2	0.659	0.627	0.994	0.999	0.497
-2,-1,-1,-1,-1	0.797	0.974	0.972	0.976	0.485
-2,-1,-1,-1,0	0.736	0.976	0.960	0.966	0.614
-2,-1,-1,-1,1	0.765	0.980	0.985	0.975	0.746
-2,-1,-1,-1,2	0.758	0.966	0.986	0.953	0.892
-2,-1,-1,0,0	0.728	0.961	0.945	0.998	0.574
-2,-1,-1,0,1	0.774	0.946	0.975	0.996	0.703
-2,-1,-1,0,2	0.750	0.979	0.952	0.995	0.820
-2,-1,-1,1,1	0.765	0.877	0.915	0.999	0.594
-2,-1,-1,1,2	0.736	0.879	0.875	1.000	0.689
-2,-1,-1,2,2	0.633	0.612	0.655	1.000	0.505
-2,-1,0,0,0	0.730	0.874	0.988	0.987	0.535
-2,-1,0,0,1	0.687	0.871	0.989	0.994	0.652
-2,-1,0,0,2	0.683	0.909	0.984	0.992	0.765
-2,-1,0,1,1	0.611	0.786	0.966	0.999	0.573
-2,-1,0,1,2	0.721	0.846	0.976	0.997	0.658
-2,-1,0,2,2	0.667	0.742	0.736	1.000	0.511
-2,-1,1,1,1	0.672	0.818	0.994	0.990	0.553
-2,-1,1,1,2	0.668	0.821	0.994	0.994	0.625
-2,-1,1,2,2	0.596	0.668	0.876	1.000	0.514
-2,-1,2,2,2	0.730	0.724	0.997	0.997	0.490
-2,0,0,0,0	0.669	0.989	0.995	0.976	0.504
-2,0,0,0,1	0.723	0.988	0.996	0.986	0.629
-2,0,0,0,2	0.616	0.990	0.989	0.989	0.747
-2,0,0,1,1	0.654	0.951	0.936	0.993	0.566
-2,0,0,1,2	0.710	0.964	0.950	0.997	0.652
-2,0,0,2,2	0.594	0.710	0.738	1.000	0.505
-2,0,1,1,1	0.668	0.929	0.996	0.993	0.528
-2,0,1,1,2	0.613	0.941	0.997	0.997	0.619

Correlation	1st Const.	2nd Const.	3rd Const.	4th Const.	5th Const.
-2,0,1,2,2	0.634	0.754	0.857	1.000	0.500
-2,0,2,2,2	0.660	0.746	0.997	0.999	0.498
-2,1,1,1,1	0.711	0.997	0.991	0.995	0.532
-2,1,1,1,2	0.773	0.994	0.987	0.986	0.610
-2,1,1,2,2	0.597	0.892	0.867	1.000	0.494
-2,1,2,2,2	0.735	0.913	0.999	0.997	0.491
-2,2,2,2,2	0.609	0.994	0.994	0.994	0.477
-1,-1,-1,-1,-1	0.966	0.990	0.961	0.981	0.441
-1,-1,-1,-1,0	0.963	0.981	0.969	0.978	0.586
-1,-1,-1,-1,1	0.976	0.972	0.969	0.979	0.759
-1,-1,-1,-1,2	0.980	0.962	0.965	0.945	0.870
-1,-1,-1,0,0	0.904	0.966	0.917	0.996	0.553
-1,-1,-1,0,1	0.938	0.959	0.934	0.997	0.678
-1,-1,-1,0,2	0.944	0.943	0.946	0.998	0.800
-1,-1,-1,1,1	0.933	0.944	0.906	0.999	0.601
-1,-1,-1,1,2	0.878	0.895	0.902	1.000	0.676
-1,-1,-1,2,2	0.639	0.685	0.669	1.000	0.499
-1,-1,0,0,0	0.836	0.865	0.996	0.993	0.529
-1,-1,0,0,1	0.876	0.916	0.996	0.986	0.666
-1,-1,0,0,2	0.891	0.870	0.999	0.983	0.765
-1,-1,0,1,1	0.859	0.848	0.938	0.998	0.591
-1,-1,0,1,2	0.819	0.849	0.949	0.999	0.662
-1,-1,0,2,2	0.595	0.582	0.712	1.000	0.491
-1,-1,1,1,1	0.800	0.795	0.991	0.995	0.543
-1,-1,1,1,2	0.857	0.882	0.990	0.995	0.626
-1,-1,1,2,2	0.676	0.659	0.879	1.000	0.491
-1,-1,2,2,2	0.686	0.704	0.991	0.999	0.486
-1,0,0,0,0	0.879	0.983	0.970	0.972	0.500
-1,0,0,0,1	0.839	0.984	0.995	0.992	0.622
-1,0,0,0,2	0.878	0.988	0.992	0.974	0.718
-1,0,0,1,1	0.807	0.984	0.932	0.998	0.565
-1,0,0,1,2	0.833	0.982	0.978	0.995	0.664
-1,0,0,2,2	0.712	0.772	0.729	0.999	0.506
-1,0,1,1,1	0.807	0.922	0.995	0.997	0.519
-1,0,1,1,2	0.848	0.937	0.992	0.995	0.638
-1,0,1,2,2	0.600	0.740	0.861	1.000	0.500
-1,0,2,2,2	0.653	0.759	0.999	0.997	0.482
-1,1,1,1,1	0.759	0.991	0.989	0.993	0.522
-1,1,1,1,2	0.819	0.996	0.991	0.987	0.612
-1,1,1,2,2	0.802	0.950	0.898	1.000	0.522
-1,1,2,2,2	0.763	0.893	0.999	0.996	0.481
-1,2,2,2,2	0.824	0.996	0.987	0.993	0.476
0,0,0,0,0	0.983	0.986	0.975	0.976	0.500
0,0,0,0,1	0.970	0.958	0.987	0.986	0.604
0,0,0,0,2	0.970	0.978	0.974	0.989	0.743
0,0,0,1,1	0.933	0.947	0.964	0.999	0.558
0,0,0,1,2	0.952	0.921	0.933	0.999	0.663
0,0,0,2,2	0.770	0.766	0.738	1.000	0.491
0,0,1,1,1	0.897	0.913	0.992	0.997	0.528
0,0,1,1,2	0.930	0.924	0.997	0.994	0.616
0,0,1,2,2	0.740	0.705	0.879	1.000	0.491
0,0,2,2,2	0.774	0.784	0.993	0.999	0.471
0,1,1,1,1	0.899	0.986	0.984	0.988	0.505
0,1,1,1,2	0.913	0.992	0.989	0.990	0.600
0,1,1,2,2	0.779	0.900	0.863	1.000	0.499
0,1,2,2,2	0.807	0.930	0.998	0.996	0.500
0,2,2,2,2	0.808	0.991	0.997	0.994	0.469
1,1,1,1,1	0.978	0.989	0.993	0.975	0.524
1,1,1,1,2	0.995	0.994	0.980	0.984	0.579
1,1,1,2,2	0.883	0.898	0.897	1.000	0.491
1,1,2,2,2	0.914	0.920	0.997	0.998	0.491
1,2,2,2,2	0.919	0.996	0.994	0.997	0.477
2,2,2,2,2	0.994	0.983	0.989	0.991	0.474

3.7 Summary and Discussion

The purpose of this chapter was to examine what makes one heuristic perform better than other heuristics. To accomplish this goal, 1120 (2KP) and 3780 (5KP) problems were examined based on problem constraint slackness setting and correlation structure. The research focused on which heuristic gave the best solution under varying conditions. The methodology included Chi-square and sign tests to prove whether or not a heuristic method was significantly better than the other methods. In the results and analyses sections, the best heuristic was examined. The chapter also studied why the best heuristic behaved as it did as a function of problem characteristics. Five heuristics, TOYODA; S – T; L – M M1; FOX; and KOCHEN, were examined under different combinations of constraint slackness and correlation structure. For equal constraint slackness settings, KOCHEN performed the best because its delayed weighting scheme is the most effective. It is very important to point out that KOCHEN gives equal weight to all constraints during the early iterations, and then places higher weight on the constraints with the least resource remaining for two reasons: to balance resource usage in equal slackness settings or to give more emphasis to the dominant constraint in mixed slackness settings. This is why KOCHEN yields better solutions for the instance where there are at least two dominant constraints. For mixed constraint slackness in the 2KP and one dominant constraint slackness setting **S** = (1, 2, 2, 2, 2) in 5KP, S – T is the best heuristic because its surplus vector, *R*, places greater weight on the tighter constraint (dominant constraint) during the entire solution process.

If correlation structures create conditions causing a dominant constraint, heuristics demonstrate similar performance characteristics as with slackness settings. For the 2KP, if ρ_{CA^1} and ρ_{CA^2} have similar values, neither constraint dominates. In this case, KOCHEN is the best heuristic for the same reason as in the equal constraint slackness settings. However, if there is only one higher positive correlation value, such as $\rho_{CA^i} = 0.9$, this correlation structure creates a dominant constraint, similar to the mixed constraint slackness setting: then FOX's absolute weighting fills the same role as S – T's surplus vector, *R*, in the mixed slackness setting. Therefore, FOX is the best heuristic when there is one dominant constraint created by the correlation structure.

In conclusion, KOCHEN is the best performing heuristic when at least two dominant constraints exist, S – T is the best heuristic when one dominant constraint is created by constraint slackness, and FOX is the best heuristic when one dominant constraint is created by the correlation structure.

IV. Empirical Analysis of Legacy Transformation Heuristics

4.1 Introduction

Legacy greedy heuristics can be viewed as path structured algorithms. A greedy heuristic makes a sequence of choices until feasibility constraints prevent further choices. However, a greedy heuristic cannot change the selection of the item after a choice is made. Local improvement procedures change current solutions to try and improve the solution.

Glover (1977) and Pirkul (1987) combine the merits of greedy heuristics and local improvement using a multiplier method and surrogate constraints to transform the MKP into a single constraint knapsack problem. A class of surrogate constraint heuristics also provides approximate, near optimal solutions to integer programming. These solutions provide a bound to the original MKP. This chapter examines the heuristic methods of Glover (1977) and Pirkul (1987) analyzed against both the 1120 problem 2KP test set and 3780 problem 5KP test set.

4.2 Background

Two transformation approaches are "surrogate relaxation" and "Lagrangian relaxation". The result is either a one dimensional knapsack problem or an unconstrained knapsack problem, respectively, both of which should be easier to solve than the original higher dimensional problem.

The following 2KP example illustrates Lagrangian relaxation:

Maximize $$Z = \sum_{j=1}^{n} c_j x_j \tag{43}$$

subject to $$\sum_{j=1}^{n} a_{1j}\, x_j \le b_1 \tag{44}$$

$$\sum_{j=1}^{n} a_{2j}\, x_j \le b_2 \tag{45}$$

$$x_j = 0 \quad \text{or} \quad 1 \qquad\qquad j = 1, 2, \ . \ . \ . \ , n \tag{46}$$

Assume that the above problem is relatively easy to solve if the second constraint is removed. The Lagrangian relaxation problem moves the second constraint into the objective function to obtain:

Maximize $$Z_{LR}(\lambda) = \sum_{j=1}^{n} c_j\, x_j + \lambda\left(b_2 - \sum_{j=1}^{n} a_{2j} x_j\right) \tag{47}$$

subject to $$\sum_{j=1}^{n} a_{1j}\, x_j \le b_1 \tag{48}$$

$$\lambda \ge 0,\ \ x_j = 0 \quad \text{or} \quad 1 \qquad\qquad j = 1, 2, \ . \ . \ . \ , n \tag{49}$$

The unconstrained objective function with binary variables follows:

$$Z_{LR}(\xi) = Max\left\{\sum_{i=1}^{2} \xi_i b_i + \sum_{j=1}^{n} x_j (c_j - \sum_{i=1}^{2} \xi_i a_{ij})\right\} \tag{50}$$

It is clear that all feasible solutions for the original MKP, Equations (43) – (46) are also feasible solutions to the Lagrangian relaxation (47) – (49) because the solution of the Lagrangian relaxation requires satisfaction of only one of the constraints. While the optimal solution of the Lagrangian relaxation (one dimensional knapsack problem) may not be feasible in the original MKP, it can provide an upper bound for the MKP. Nemhauser and Wolsey (1988) suggest that Lagrangian relaxation provides tighter bounds to the IP than the linear programming relaxation meaning $Z_{IP} \le Z_{LR}(\lambda^*) \le Z_{LP}$, where λ^* is the optimal Lagrangian multipliers found by solving $Z_{LR}(\lambda^*) = \min_{\lambda \geq 0} Z_{LR}(\lambda)$

and Z_{LP} is the optimal objective function value of the linear programming relaxation.

In surrogate relaxation, the original set of constraints are replaced with a single constraint, a nonnegative linear combination of the original constraints. Glover (1968) defined the surrogate problem as follows:

$$\begin{aligned} & Z_S(\mu) = \max\ c\,x \\ \text{subject to} \quad & (\mu A)\,x \leq \mu b \\ & x_j \in \{0,1\} \ \ \forall \ j \in N \end{aligned} \tag{51}$$

where μ is a positive multiplier vector of size *m*. The formulation of the surrogate relaxation of the 2KP of (43) – (46) is as follows:

$$\text{Maximize} \qquad Z_S(\mu) = \sum_{j=1}^{n} c_j\, x_j \tag{52}$$

$$\text{subject to} \qquad \mu_1 \sum_{j=1}^{n} a_{1j}\, x_j + \mu_2 \sum_{j=1}^{n} a_{2j} x_j \leq \mu_1\, b_1 + \mu_2 b_2 \tag{53}$$

$$\mu_i \geq 0, \ \ x_j = 0 \ \text{ or } \ 1 \qquad i = 1, 2 \quad j = 1, 2, \ . \ . \ . \ , n \tag{54}$$

The solution of the surrogate relaxation also produces an upper bound on the MKP optimal solution. The best bound using this scheme is determined by locating a set of multipliers μ^* such that

$$Z_S(\mu^*) = \underset{\mu}{Min}\ \ Z_S(\mu) \tag{55}$$

If μ^* is known, then $Z_S(\mu^*)$ provides better bounds than both the LP-relaxation and the Lagrangian relaxation (Glover, 1968). The surrogate relaxation problem of Equations (43) through (46) is a 0 – 1 knapsack problem, which is easier to solve than the original MKP.

4.3 Glover's Surrogate Constraint Heuristic

Glover (1977) proposes a heuristic using surrogate relaxation for obtaining near-optimal solutions to integer programming problems. Glover introduces the framework for the surrogate constraint heuristic, defined as a sequence of four steps:

Step 1: Generate the surrogate constraint using dual variables of the relaxed problem

Step 2: Determine a feasible starting solution.

Step 3: By reference to the surrogate constraint, periodically or regularly updated, establish measures of the goodness of increasing and decreasing the value of each variable.

Step 4: Sequentially change – increase or decrease – the values of the variables, singly or in blocks, in accordance with their goodness measure, and keep track of the best solution found in the process.

Glover also suggests temporarily allowing infeasible solutions by the use of an oscillating assignment heuristic to solve the 0 – 1 surrogate KP. The oscillating assignment heuristic is a combination of the dual gradient approach of Senju and Toyoda (1968), and a primal gradient approach, (Kochenberger *et al.*,1974). The oscillating assignment heuristic integrates these two different gradient approaches depending on whether or not the current solution is feasible. When a current solution is infeasible, remove variables (set them equal to zero) until feasibility is achieved. By contrast, variables are forced to one to move from the feasible region into the infeasible region. Details of the oscillating assignment heuristic are summarized as follows (Glover, 1977):

Step 1: Form the KP using surrogate multipliers

Index variables by decreasing $\frac{c_j}{(\mu A)_j}$

Fix variables equal to one according to the order; if fixing a variable equal to one violates any of the constraints, fix that variable equal to zero and continue

Define X^0 as the current solution.

Define $S1=\{j \mid x_j^0 = 1\}$ and $S0=\{j \mid x_j^0 = 0\}$

$SM1= S1\backslash\{\text{most recent move into } S1\}$

$SM0= S0\backslash\{\text{most recent move into } S0\}$

Step 2: Generate next solution:

$L_0 = L_1 = 0$

$U_0 = \infty$ or average $\{SM0\}$

if smallest index move creates an infeasible solution

$U_1 = \infty$ or average $\{SM1\}$

if largest index move creates a feasible solution

If X^0 feasible, $S1 \cup \arg\min\{j\in SM0 \mid L_0 \le a_j \le U_0\}$

If X^0 infeasible, $S0 \cup \arg\max\{j\in SM1 \mid L_1 \le a_j \le U_1\}$

Step 3: Generate a trial solution:

If X^0 feasible then: temporarily transfer arg max$\{c_j \mid j\in SM0$ and transfer feasible$\}$ to $S1$. Repeat until feasibility destroyed by any move

If X^0 infeasible then: temporarily transfer arg min$\{c_j \mid j\in SM1$ and transfer feasible$\}$ to $S0$. If single transfer does not yield a feasible solution, select smallest c_j and repeat. If ties occur for smallest c_j pick largest index.

Once feasibility attained apply X^0 feasible rule above

Step 4: Max iterations reached then STOP; Else go to Step 2.

Glover's oscillating assignment heuristic (GLOVER) can be applied using dual variables easily acquired from the LP relaxation. Glover's oscillating assignment heuristic is programmed in various iteration counts [1, 2, 3, 4, 5, 10, 15, 20, 25, 50, 100, 150 and 200] to evaluate the solution performance of the GLOVER. Glover's heuristic performance is examined using average relative error.

This oscillating assignment approach is an early basis for tabu search's strategic oscillation feature. In summary, when proceeding from inside the feasible region to a point where no improving move exists, except one that violates feasibility, the heuristic follows a path that goes outside the feasible region. By contrast, proceeding from an infeasible point discards the stopping rule of the dual approach upon entering the feasible region, in favor of continuing to go deeper along a path dictated by the surrogate constraints. Thereupon, the procedure reverses, working back toward the periphery, taking a trajectory designed to obtain improving solutions. Figure 17 plots the change in average relative error as a function of the number of iterations in all problems of the 2KP test set (four slackness settings).

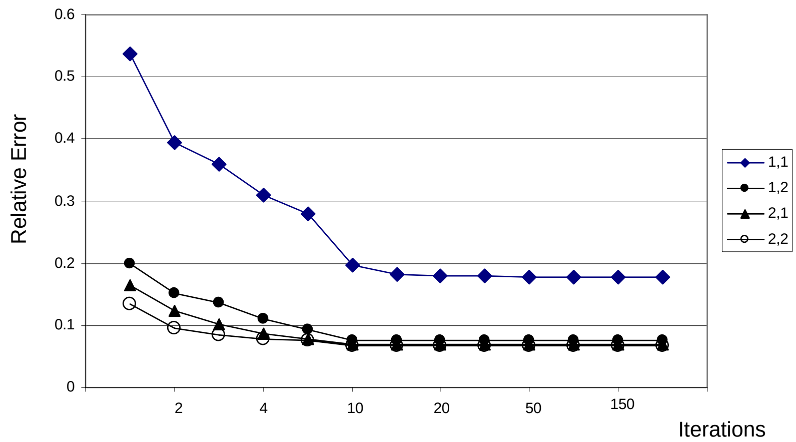


Figure 17. Relative Error Convergence according to Number of Iterations

Figure 17 shows how much Glover's local improvement scheme decreases average relative error (improves solution) as one increases the number of iterations, plotted by the four slackness settings. Figure 17 indicates that Glover's oscillating assignment solution converges to its best solution after 50 iterations. For slackness, **S** = (1, 1), the average relative error in the first iteration is 0.54 while the average relative error in the 50th iteration is 0.18, an improvement of 36%.

The results reported in Table 27 show the average relative errors by Glover's oscillating assignment heuristic (GLOVER) for iteration 1 and iteration 50, and by the best heuristic among the five legacy greedy heuristics, TOYODA, S – T, L – M M1, FOX, or KOCHEN.

Table 27. Average Relative Error of GLOVER vs. Best Legacy

	Slackness	GLOVER		Best Legacy
		1 iteration	50 iterations	
2KP	(1, 1)	0.538	0.175	0.654
	(1, 2)	0.198	0.075	0.167
	(2, 1)	0.166	0.069	0.132
	(2, 2)	0.133	0.062	0.139
5KP	(1,1,1,1,1)	1.123	0.469	2.804
	(1,1,1,1,2)	0.865	0.335	1.689
	(1,1,1,2,2)	0.635	0.221	1.034
	(1,1,2,2,2)	0.477	0.147	0.588
	(1,2,2,2,2)	0.277	0.105	0.289
	(2,2,2,2,2)	0.231	0.114	0.539

(Unit: Percent)

Table 27 indicates that GLOVER with 50 iterations yields better solutions than the best solution found by any of the legacy heuristics, TOYODA, S – T, L – M M1, FOX and KOCHEN. In 88.7 % (1089 of 1120 (2KP) and 3256 of 3780 (5KP) test problems) of the problems Glover's heuristic solution, with 50 iterations, is within half a percent of the optimum. In addition, GLOVER found the optimum solution in 518 of 1120 2KP problems and 1072 of 3780 5KP problems. However, GLOVER should not be compared directly with legacy heuristics because GLOVER includes the improvement phase. GLOVER first gets a feasible solution using a greedy method, and then conducts the oscillating assignment procedure to improve the quality of the solutions. Legacy heuristics do not include an improvement phase.

The performance of GLOVER is affected by constraint slackness settings. Table 27 shows that GLOVER works better on loose slackness settings than on tight slackness settings. This is because the algorithm generates more choices of variables to be selected when all constraints are loose. GLOVER works better on negative correlation structures than on positive correlation structures because positive correlation structures restrict resources as do tight constraint slackness settings.

Most importantly, the results on the 1st iteration by GLOVER have better or similar values to the results by the best legacy heuristic. The 1st iteration of GLOVER is a greedy heuristic solution. The 1st iteration of GLOVER changes MKP into KP using surrogate multipliers, and then picks items following a decreasing order of bang-for-buck ratios (benefit/cost) while maintaining feasibility. Dual variable values in the solution of LP relaxation are used as the surrogate multipliers. Dual variables are an estimate of how critical a resource is to a problem, which is related to constraint slackness and correlation structure. The dual variables (surrogate multipliers) can be said to include information regarding constraint slackness and correlation structure. For example, in **S** = (2, 1), the loose constraint is a non-binding constraint, so the dual variable of this constraint equals zero. Thus, GLOVER's 1st iteration focuses on the dominant constraint (tighter constraint) using surrogate multipliers as the best legacy heuristic uses its penalty cost function. The results for GLOVER's 1st iteration support the empirical analysis of legacy greedy heuristics in Chapter III where the best heuristic focused on the dominant constraints.

4.4 Pirkul's MKHEUR Heuristic

Pirkul (1987) introduces a local improvement search using surrogate multipliers similar to those in Glover's heuristic. Pirkul used the dual variables associated with each constraint in the linear programming relaxation as surrogate multipliers. Pirkul's heuristic procedure, MKHEUR, uses a surrogate constraint and benefit/cost ratios to determine which variables are fixed equal to one in the first iteration. The improvement phase then systematically swaps values of the selected variables to improve the current solution. Details of MKHEUR are as follows:

Step 1: Determine a set of surrogate multipliers using dual variables from the linear programming relaxation.

Step 2: Calculate $\frac{c_j}{(\mu A)_j}$ ratios, sort and renumber the variables according to the decreasing value of these ratios.

Step 3: (*a*). Fix variables equal to one according to the order determined in *Step* 2.

(*b*). If fixing a variable equal to one causes violation of one of the original constraints, fix that variable equal to zero and continue. Denote the feasible solution determined in this step as $\bar{x}$.

Step 4: For each variable fixed equal to one in $\bar{x}$, fix the variable equal to zero and repeat Step 3 (*b*) to define a new feasible solution (The number of new feasible solutions are equal to the number of variables fixed equal to one in $\bar{x}$).

Step 5: Return the best solution (the largest objective function value) found.

Step 3 of MKHEUR resembles GLOVER's 1st iteration. Since a number of different feasible solutions are identified in *Step* 4 depending on the number of variables equal to one, MKHEUR considerably improves the quality of the solution in *Step* 4 by finding the best solution when comparing all feasible solutions to each other. Procedure MKHEUR is based on a local search procedure superimposed on a greedy approach.

A set of computational experiments was conducted to evaluate the performance of MKHEUR. Table 28 shows the number of comparisons conducted by MKHEUR according to 2KP constraint slackness setting, and the relative improvement from the first feasible solution to the final solution (the minimum relative error found among all feasible solutions). Note that the number of comparisons is equal to the number of variables fixed equal to one in $\bar{x}$, *Step* 4 of MKHEUR.

Table 28. Local Improvement by Pirkul MKHEUR

Slackness	Average # of Comparisons	Initial Relative Error (Average)	Final Relative Error (Average)	Improvement
(1, 1)	38.4	0.538	0.090	0.448
(1, 2)	46.6	0.198	0.032	0.166
(2, 1)	45.8	0.166	0.022	0.144
(2, 2)	72.5	0.133	0.018	0.115

Table 28 shows a local improvement from the first solution to the best solution. The number of comparisons increases by slackness setting (1, 1) to (2, 2). This is reasonable as loose constraints admit more variables with a value equal to one. Thus, slackness **S** = (2, 2) should have the maximum number of variables equal to one, after the first iteration. MKHEUR is affected by various constraint slackness and correlation structure settings. MKHEUR works better on loose slackness settings than on tight slackness settings. There are fewer variables set to one in the presence of tight constraints, so an improper choice of variables leads to a larger relative error in tight constraint settings. Overall, the improvement phase of MKHEUR improves the objective function value by approximately 0.22. The detailed results for various correlation structures by MKHEUR are shown in Tables D.1 and D.2 in Appendix D.

Table 29. Average Relative Errors of Pirkul's Heuristic vs. GLOVER and Best Legacy

	Slackness	Pirkul MKHEUR	GLOVER (50 iterations)	Best Legacy
2KP	(1, 1)	0.090	0.175	0.654
	(1, 2)	0.032	0.075	0.167
	(2, 1)	0.022	0.069	0.132
	(2, 2)	0.018	0.062	0.139
5KP	(1,1,1,1,1)	0.264	0.469	2.804
	(1,1,1,1,2)	0.192	0.335	1.689
	(1,1,1,2,2)	0.142	0.221	1.034
	(1,1,2,2,2)	0.094	0.147	0.588
	(1,2,2,2,2)	0.063	0.105	0.289
	(2,2,2,2,2)	0.045	0.114	0.539

(Unit: Percent)

Table 29 compares MKHEUR, GLOVER and the best legacy heuristic. The results in Table 29 indicate that the procedure MKHEUR is most effective in solving problems with different slackness structures. Thus, the improvement phase of MKHEUR appears better than the oscillating assignment procedure (improvement phase) of GLOVER. In 95.7 % of problems, the Pirkul heuristic solution is within half a percent of the optimum, while GLOVER's level was 88.7%. MKHEUR found the optimum solution in 777 of the 1120 2KP problems and 1502 of the 3780 5KP problems.

Why is MKHEUR better than GLOVER? The maximum number of iterations of MKHEUR is 83 iterations for 2KP problems while GLOVER conducts 50 iterations for the same problems. Glover suggests conducting a larger number of iterations to identify a large set of different feasible solutions. However, the GLOVER heuristic repeats solutions. Thus, even if GLOVER has many iterations, the number of unique solutions identified is restricted. On the contrary, MKHEUR creates a different solution at every iteration. Thus, MKHEUR examines more solutions than GLOVER.

Pirkul points out that the effectiveness of MKHEUR is dependent on the ability of the surrogate constraint to capture aggregate consumption levels of the resources. Table 29 supports that the dual variables provided by the linear programming relaxation are very effective as surrogate multipliers. Although MKHEUR has the maximum relative error when all constraints are tight, **S** = (1, 1, 1, 1, 1), the average objective function value is still 99.74 % of optimum.

4.5 Summary

Two important findings are presented in this chapter. One finding is that an improvement phase is very effective and necessary to increase the objective function value after a simple greedy heuristic is applied. GLOVER improved the relative error by 0.32 from its first relative error 0.52 to final relative error 0.20, and MKHEUR improved the relative error by 0.41 from its first relative error 0.52 to final relative error 0.11. Even though a greedy heuristic yields an objective function value very close to the optimum, simple perturbations of solution can increase the objective function value.

The other finding is that the associated dual variables of the linear programming relaxation have information regarding constraint slackness and correlation structures. Thus, dual variables are very useful as surrogate multipliers. Even if GLOVER and MKHEUR do not conduct an improvement phase, their first solutions are sometimes higher than the best solution of legacy greedy heuristics because dual variables can function as effective penalty weights of greedy heuristics. Since the KP is easier to solve than a MKP, dual variables are effective in transforming MKP to KP.

V. New Heuristics Development Based on Empirical Analysis

5.1 Introduction

This chapter presents several new greedy heuristic approaches exploiting previous empirical study insights. Three new types of heuristic approaches are introduced. First, a typed heuristic is developed based on pre-processing a particular problem and using problem-specific knowledge to gain computational efficiencies. Second, new gradient heuristics are developed by combining the characteristics deemed important for a best performance. Third, new reduction heuristics are developed by additionally combining advantageous characteristics from the transformation heuristics.

5.2 A Typed Heuristic

The typed heuristic (TYPE) involves pre-processing a problem and picking a likely "best performer." Both Hooker (1994) and Loulou and Michaelides (1979) suggest making heuristic choices based on computed problem characteristics. Loulou and Michaelides (1979) term such an approach a typology approach. Pre-processing a combinatorial problem involves relatively little computational overhead. TYPE pre-processes a problem to determine problem characteristics (constraint slackness, problem size, and correlation structure), and chooses a heuristic most likely to produce a best solution among the suite of heuristics, TOYODA, S – T, the four L – M (M1, M2, SW1, SW2), FOX, and KOCHEN, considered.

Based on the previous empirical analysis, the best heuristic among five different heuristics, TOYODA, S – T, L – M M1, FOX, and KOCHEN was determined based on constraint slackness and correlation structure.

Tables 30 reviews constraint slackness, correlation structures, and the best heuristics under 2KP problem set results and Table 31 reviews the 5KP problem set results.

Table 30. 2KP Constraint Slackness, Correlation Structure and Best Heuristic

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="4">SLACKNESS</td></tr>
<tr><td>(1, 1)</td><td>(1, 2)</td><td>(2, 1)</td><td>(2, 2)</td></tr>
<tr><td rowspan="4">CORRELATION</td><td>(2, 2, 2)</td><td>KOCHEN</td><td>FOX</td><td>FOX</td><td>KOCHEN</td></tr>
<tr><td>(2, 1, 1)
(2, 0, 0)
(2, -1, -1)
(2, -2, -2)</td><td>FOX</td><td>FOX</td><td>KOCHEN</td><td>FOX</td></tr>
<tr><td>(1, 2, 1)
(0, 2, 0)
(-1, 2, -1)
(-2, 2, -2)</td><td>FOX</td><td>KOCHEN</td><td>FOX</td><td>FOX</td></tr>
<tr><td>Other Correlation Structures</td><td>KOCHEN</td><td>S – T</td><td>S – T</td><td>KOCHEN</td></tr>
</table>

Based on Table 30, KOCHEN is applied for equal constraint slackness while S – T is applied for mixed constraint slackness if there are not extreme levels of correlation. However, when considering the correlation structure, FOX is applied for $CA^1 = 2$ (ρ_{CA^1} =0.999) or $CA^2 = 2$ ($\rho_{CA^2} = 0.999$) for equal constraint slackness settings, or for a combination of high positive correlation and a tight constraint. When a high positive correlation is associated with a loose constraint in the mixed slackness settings, both the tight and the loose constraints behave similarly, thus KOCHEN is applied. In other words, extreme levels of correlation trump heuristic choice based on constraint slackness.

Table 31. 5KP Constraint Slackness, Correlation Structures and Best Heuristic

		SLACKNESS					
		(1,1,1,1,1)	(1,1,1,1,2)	(1,1,1,2,2)	(1,1,2,2,2)	(1,2,2,2,2)	(2,2,2,2,2)
CORRELATION	(-2, -2, -2, -2, 2) (-2, -2, -2, -1, 2) (-2, -2, -1, -1, 2) (-2, -1, -1, -1, 2) (-1, -1, -1, -1, 2)	FOX	KOCHEN	KOCHEN	KOCHEN	KOCHEN	FOX
	(-2, -2, -2, 2, 2) (-2, -2, -1, 2, 2) (-2, -1, -1, 2, 2) (-1, -1, -1, 2, 2)	KOCHEN	FOX	KOCHEN	KOCHEN	KOCHEN	KOCHEN
	(-2, -2, 2, 2,2) (-2, -1, 2, 2,2) (-1, -1, 2, 2,2)	KOCHEN	KOCHEN	FOX	KOCHEN	KOCHEN	KOCHEN
	(-2, 2, 2, 2, 2) (-1, 2, 2, 2, 2)	KOCHEN	KOCHEN	KOCHEN	FOX	KOCHEN	KOCHEN
	(2, 2, 2, 2, 2)	KOCHEN	KOCHEN	KOCHEN	KOCHEN	FOX	KOCHEN
	Other Correlation Structures	KOCHEN	KOCHEN	KOCHEN	KOCHEN	S – T	KOCHEN

Strategies based on Table 31 agree with the 2KP strategies based on Table 30. This typology was coded and run against the 2KP and 5KP problem sets. The results of TYPE are presented below. Note, TYPE does not run all heuristics and return the best solution, but selects a single heuristic to run based on the generalized rules for selection. Figure 18 shows the type rule set employed and Figure 19 shows the flowchart of the TYPE heuristic.

```
If all slackness are equal then
      If only one correlation with objective function
         is strongly positive then

                  Run FOX
      Else
                  Run KOCHEN

Else If at least two slackness are tight then
      If only one correlation with objective function
         is strongly positive under the tight slackness then

                  Run FOX
      Else
                  Run KOCHEN


Else If only one constraint slackness is tight then
      If only one correlation with objective function
         is strongly positive under the tight slackness then

                  Run FOX

      Else If correlation with objective function
         is strongly positive under the loose slackness then

                  Run KOCHEN
      Else

                  Run S – T
```

Figure 18. Pseudo-Code That Lays Out Type Rule Set Employed

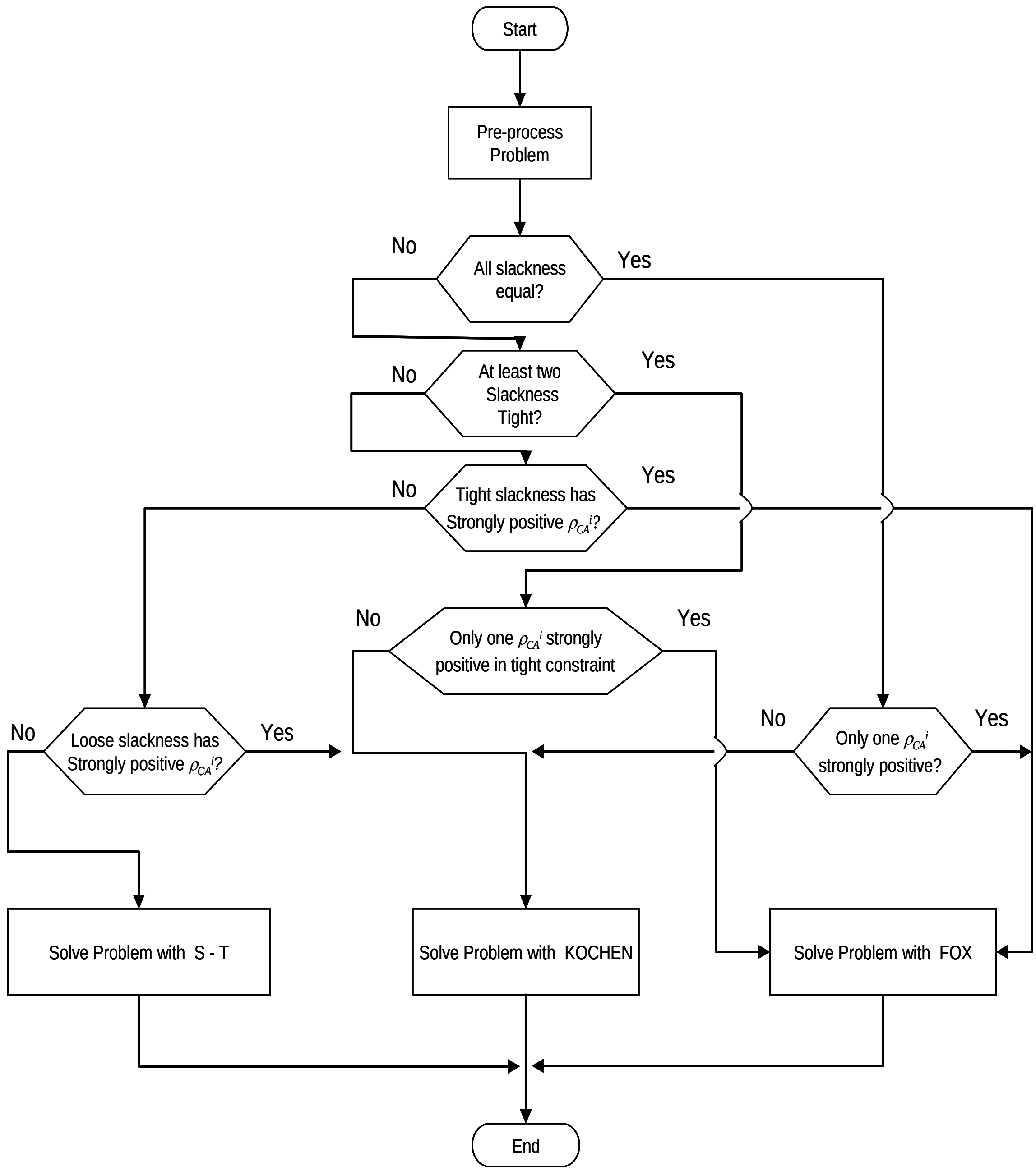


Figure 19. Flowchart of TYPE heuristic

Analysis of Results of TYPE Heuristic

The performance of TYPE heuristic by constraint slackness is summarized in Table 32 and graphed in Figure 20. Unlike previous data, this data includes ties to demonstrate overall TYPE performance compared to the legacy heuristics.

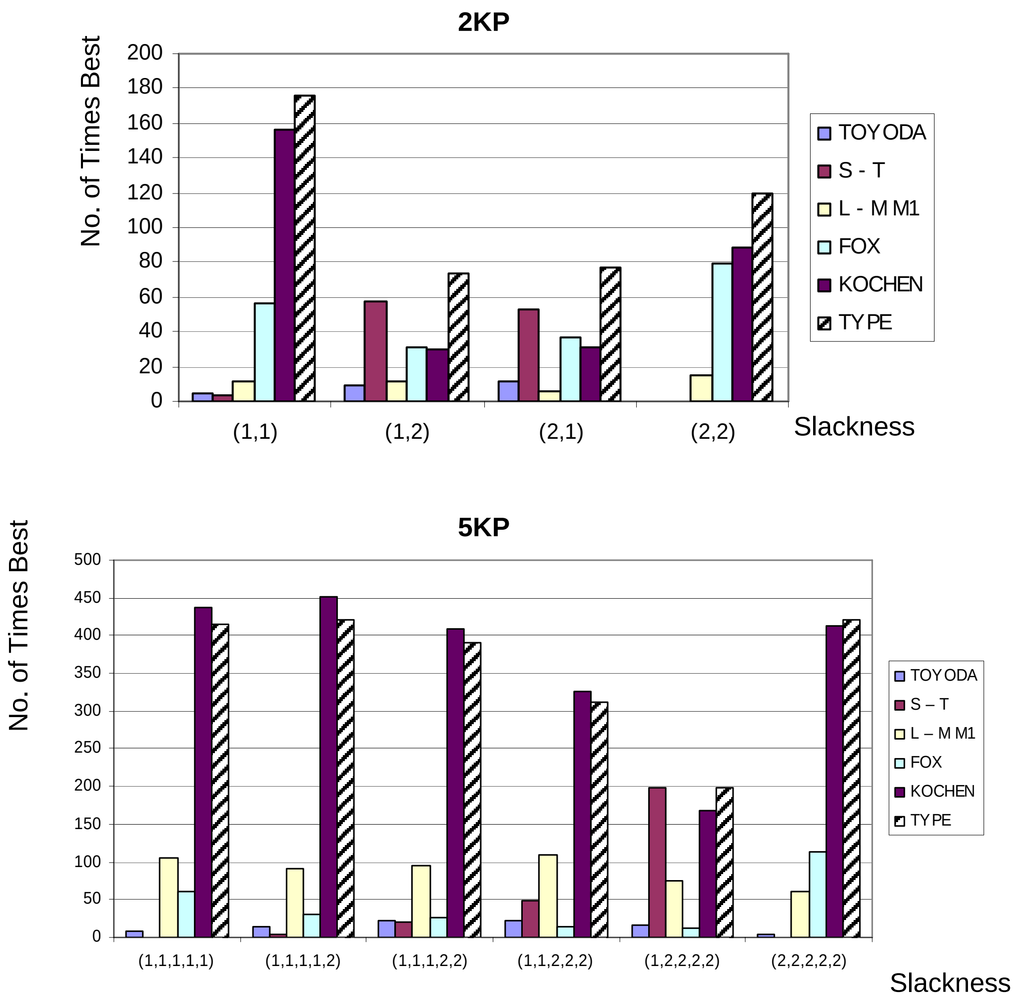


Figure 20. Comparison of TYPE Heuristic under Various Constraint Slackness

Table 32. Number of Times Equal to Best by TYPE under Constraint Slackness

	Slackness	**TYPE**	TOYODA	S – T	L – M M1	FOX	KOCHEN
2KP	(1, 1)	**176**	5	3	11	56	156
	(1, 2)	**74**	9	58	11	31	30
	(2, 1)	**77**	11	53	6	37	31
	(2, 2)	**119**	0	0	15	79	88
5KP	(1,1,1,1,1)	**415**	8	1	105	61	437
	(1,1,1,1,2)	**422**	15	4	92	31	452
	(1,1,1,2,2)	**390**	22	20	96	26	408
	(1,1,2,2,2)	**312**	22	49	109	15	326
	(1,2,2,2,2)	**199**	16	198	74	12	168
	(2,2,2,2,2)	**421**	5	1	60	114	412

The TYPE heuristic has slightly better or similar performance than the prior best heuristic, but more importantly, overall consistent performance for various constraint slackness settings. As the problem sets consider a full range of correlation structures, confidence is high that demonstrated results can be achieved for all MKP.

The TYPE heuristic performs particularly well against correlation structure in both the 2KP and the 5KP problems. Among 45 feasible correlation structures of 2KP, the TYPE heuristic is the best (including ties) in 32 correlation structures. The TYPE heuristic is the best performer for 69 out of 126 correlation structures in 5KP. The key trend is the consistent level of performance of the TYPE heuristic. The detailed data, breaking performance out by each of the correlation structures in the 2KP and 5KP problem sets, is provided in Tables E.1 and E.2, respectively, in Appendix E.

Table 33 shows, from a different comparative point of view, the comparison between the TYPE heuristic and each legacy greedy heuristic in terms of percentage of solutions better than, equal to, or worse than a specified legacy heuristic over the 2KP and 5KP problem sets combined.

Table 33. Comparison of TYPE Heuristic with Test Heuristics

TYPE vs.	TYPE Better	TYPE Same	TYPE Worse
TOYODA	84.7	9.3	6.0
S–T	70.4	22.6	7.1
L–M M1	78.8	4.0	17.3
FOX	78.0	16.7	5.3
KOCHEN	10.6	79.4	9.9

(Unit: Percent)

Table 33 results indicate that the TYPE heuristic meets or exceeds the performance of the greedy legacy heuristics an average of 90.9% of the time. The TYPE approach provides improved performance over any particular heuristic both in terms of obtaining best solutions and reducing the occurrence of worst solutions. Clearly, the TYPE heuristic performs very well.

5.3 New Gradient Heuristic

All heuristics use penalty factors to consider constraints. The previous empirical studies uncovered influential factors in heuristic performance. New gradient heuristics are developed that include these influential factors. This section presents three new gradient heuristic approaches that exploit the insights of the empirical study. These heuristics are labeled as new gradient heuristics version 1 through version 3 (NG V1, NG V2, NG V3) according to the chronological order of development. NG V1 heuristic improves the solution trajectory through the feasible region. NG V2 heuristic modifies the delayed weighting scheme of KOCHEN to be suitable for any constraint slackness settings. NG V3 is the final new gradient heuristic and combines the merit of NG V1 and NG V2. NG V3 improves the effective gradient function using the lognormal distribution in order to respond well to various combinations of constraint slackness setting and correlation structures.

5.3.1 New Gradient Heuristic Version 1 (NG V1)

The first new primal effective gradient heuristic is NG V1. KOCHEN was the best performer for equal constraint slackness except when correlation structures contained strong positive correlation values. In this case, FOX was the best performer. The dual effective gradient method, S – T, was the best when the constraint slackness levels were mixed in the 2KP or slackness settings (1, 2, 2, 2, 2) in the 5KP. The characteristics of KOCHEN and S – T are combined with the FOX heuristic to create a primal heuristic with an improved trajectory through the feasible region.

The following algorithm is a general explanation of NG V1. NG V1 differs in defining the penalty cost function in step 3 of TOYODA. Using the notation of Toyoda (1975): I_j = item j, j = 1, …, n; T = Set of all items; Tu = set of items accepted so far; T_D = set of items not in Tu, $T \setminus Tu$ (*i.e.*, $I_j \notin Tu \Rightarrow I_j \in T_D$); Tc = set of candidate items; C_i = total resource required by the set of accepted items in ith constraint, *i.e.*, $C_i = \sum_{I_j \in Tu} a_{ij}$;

Z = objective function value; $\mathbf{P_U}$ is the cumulative total resource used vector where $\mathbf{P_U}$ = (C_1, …, C_m); r_i is the surplus resource of each constraint, surplus vector, **R,** as used in S – T heuristic, *i.e.*, $\mathbf{R} = (r_1, \ldots, r_m)$, $r_i = \sum_{j \in N} a_{ij} - b_i$; **S** is the slackness ratio vector of the right hand side of constraint i to the sum of the coefficients in that constraint, *i.e.*,

$S_i = \dfrac{b_i}{\sum_{j \in N} a_{ij}}$; vector of RHS values, $\mathbf{B} = (b_1, \ldots, b_m)$; $\mathbf{A_j}$ = vector of constraint coefficients of variable j, $\mathbf{A_j} = (a_{1j}, \ldots, a_{mj})$; the value of item (objective function coefficient) is c_j; $x_j \in \{0, 1\}$.

Step 1: Initialization.

$Tu = \varnothing$, $T_D = T$, $\mathbf{P_U} = (0, 0)$

$Z = 0$, $x_i = 0$, $C_i = 0$, $i = 1, \ldots, m$, $j = 1, \ldots, n$

Step 2: Assign all candidate items to Tc, candidate item set.

$$Tc = \{I_j \mid I_j \in T_D \ \ and \ \ A_j \le \mathbf{B} - \mathbf{P_U}\}$$

If $Tc = \varnothing$ STOP; Otherwise go to *Step 3.*

Step 3: Compute effective gradient for items in Tc as follows:

(a) If $\mathbf{P_U}$ is a zero vector then: $G_j = \dfrac{c_j}{\sum_{i=1}^{m} r_i \cdot a_{ij}}$

(b) Otherwise, compute for j, $j = 1, \ldots, n$

$$q = \arg\min_i \{(S_i) \cdot (1 - C_i - a_{ij})\} \text{ for all } i$$

$$v_j = \frac{(1 - S_q) \cdot a_{qj}}{(1 - C_q) + \sum_{i=1, i \ne q}^{m} a_{ij} \cdot [Exp(C_i)] \cdot (1 - S_i)}$$

$$G_j = \frac{c_j}{v_j}$$

Step 4: $k = \arg\max \{ G_j \mid I_j \in Tc$ and feasible $\}$

Step 5: Calculate: $Z \leftarrow Z + c_k$, $X_k = 1$, $Tu \leftarrow Tu \cup \{I_k\}$

$\mathbf{P_U} \leftarrow \mathbf{P_U} + A_k$, $C_i \leftarrow C_i + a_{ik}$ for all i, $T_D = T \setminus \{I_k\}$

Go to *Step 2.*

Figure 21 shows the flowchart of the NG V1 heuristic procedure.

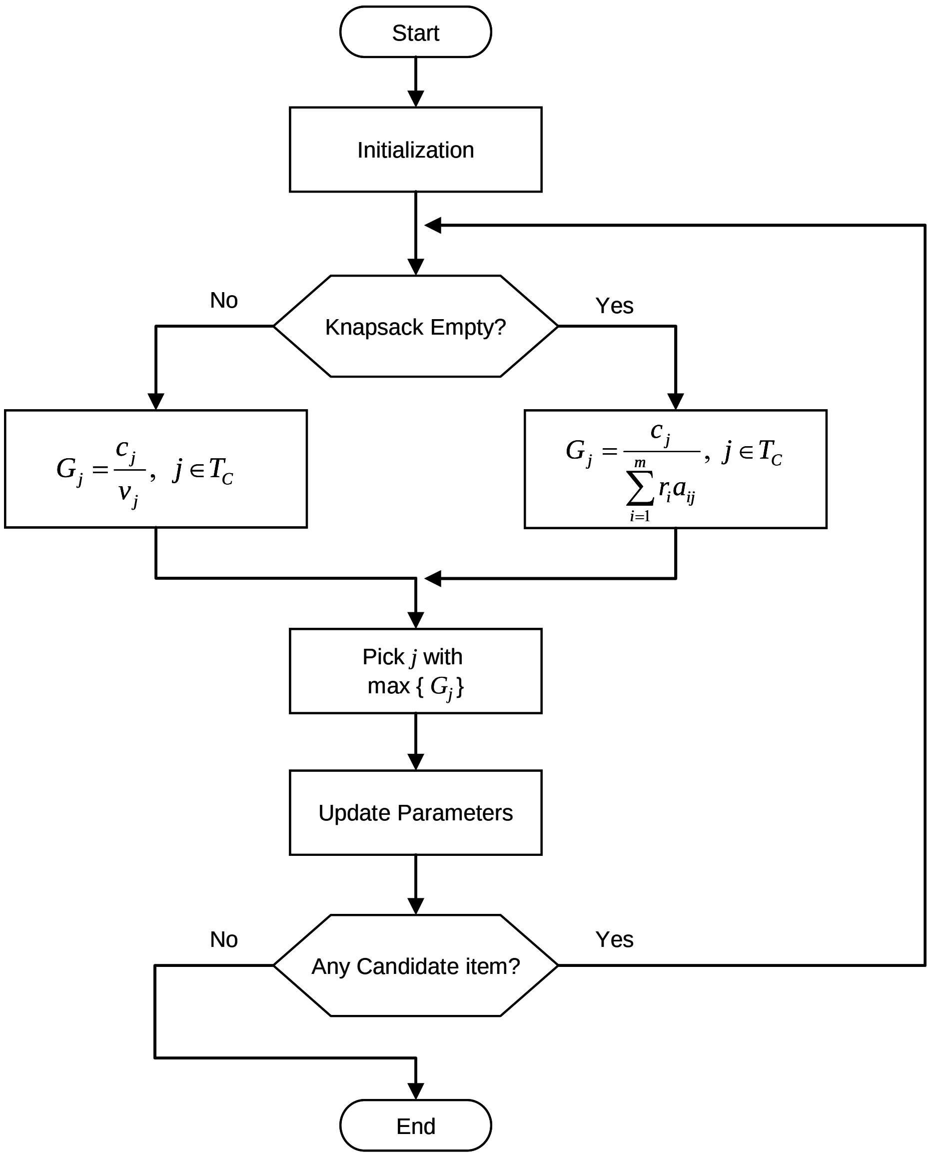


Figure 21. Flowchart of NG V1 Heuristic

Each iteration selects a new item with the largest effective gradient. NG V1 combines merits of three legacy heuristics: The first merit is the surplus vector, **R**, of the dual heuristic, S – T. In *Step* 3, multiplying the surplus vector, r_i, and constraint coefficients, a_{ij}, modifies the direction in the feasible space from the initial iteration especially when two constraints differ greatly. The second merit is the absolute weighting scheme of FOX and L – M M1, which was suitable for strong positive correlation among coefficients. The absolute weighting scheme focuses on a dominant constraint. The multiplier C_i considers the cumulative amount added to the *i*th constraint. Choosing the constraint with the minimum value of $S_i \cdot (1 - C_i - a_{ij})$ plays a similar role to the absolute weighting scheme. The value of $S_i \cdot (1 - C_i - a_{ij})$ represents the amount of resource remaining in constraint *i* if item *j* is added to the current solution (set $x_j = 1$), which indicates the constraint with the least resource remaining. The third merit is the delayed weighting scheme used in KOCHEN. This weighting scheme gives similar weights to all constraints in the early iterations, while heavily weighting the dominant constraint in the later iterations. Multiplying constraint coefficients by $\frac{1}{1-C_i}$ rather than $Exp(C_i)$ causes consideration of the more dominant constraint. The values of $\frac{1}{1-C_i}$ and $Exp(C_i)$ give nearly equal weights on all constraints until late iterations, and then the value of $\frac{1}{1-C_i}$ gives more weight to the dominant constraint (the constraint having the least resource remaining).

For example, in a 2KP, if the characteristic of a problem is a tight first constraint and a loose second constraint, r_1 should be large and r_2 should be small in the first iteration. This allows NG V1 to pick an item with a small coefficient in the tight constraint. As the number of iterations increases, the value of $\frac{1}{1-C_i}$ grows larger than the value of $Exp(C_2)$. With this weighting scheme, NG V1 focuses on the tight constraint. NG V1 works like S – T but varies weights to the dominant constraint, while S – T gives constant weight to all constraints. This weighting scheme provides a better direction by considering the tight constraint. In other words, these three merits of NG V1 provide a direction through the feasible region, a direction biased toward the dominant constraint, as desired.

Analysis of the results of NG V1 Heuristic

The deflection of the greedy heuristic solution direction is obtained by weighting each penalty term by the constraint slackness levels. Consider a 2-dimensional plot of constraint slackness levels. The figure formed by the regions from 0 to the right hand side value of each constraint on either axis forms a rectangular region of feasible solutions. When constraints are similar, this figure is a square. The S – T heuristic works best on the mixed constraints as the plot of solutions (each iteration) forms a direction favoring the corner of the rectangle (See Figure 8). The solution sequence of most primal methods cuts through the feasible region at a 45 degree angle. NG V1 adjusts its direction toward the rectangle corner point. NG V1 was compared to the five legacy greedy heuristics over all 2KP and 5KP test problems with the results provided in Table 34.

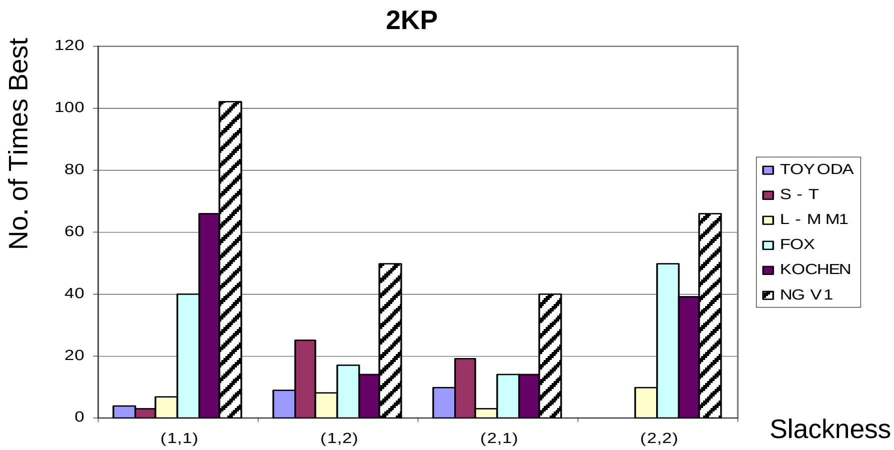
Table 34. Comparison of NG V1 Heuristic with Test Heuristics

NG V1 vs.	NG V1 Better	NG V1 Same	NG V1 Worse
TOYODA	85.9	4.0	10.1
S – T	78.5	10.2	11.3
L – M M1	83.0	5.9	11.0
FOX	86.3	6.1	7.5
KOCHEN	47.2	15.9	37.0

(Unit: Percent)

Table 34 shows that NG V1 meets or exceeds the performance of TOYODA 90% of the time, S – T 89% of the time, L – M M1 89% of the time, FOX 92% of the time, and KOCHEN 63% of the time. These results are particularly appealing since NG V1 heuristic is a general purpose greedy approach not tied to pre-processing a problem. NG V1 provides relatively robust performance results against problems having the diversity of problems expected in actual practice.

Figure 22 and Table 35 provide the results of competitively testing NG V1 heuristic against the legacy heuristics in terms of number of best solutions by slackness settings. Ties are not counted in the data presented in Figure 22 and Table 35. Table E.3 of Appendix E provides the detailed data of the sign test.



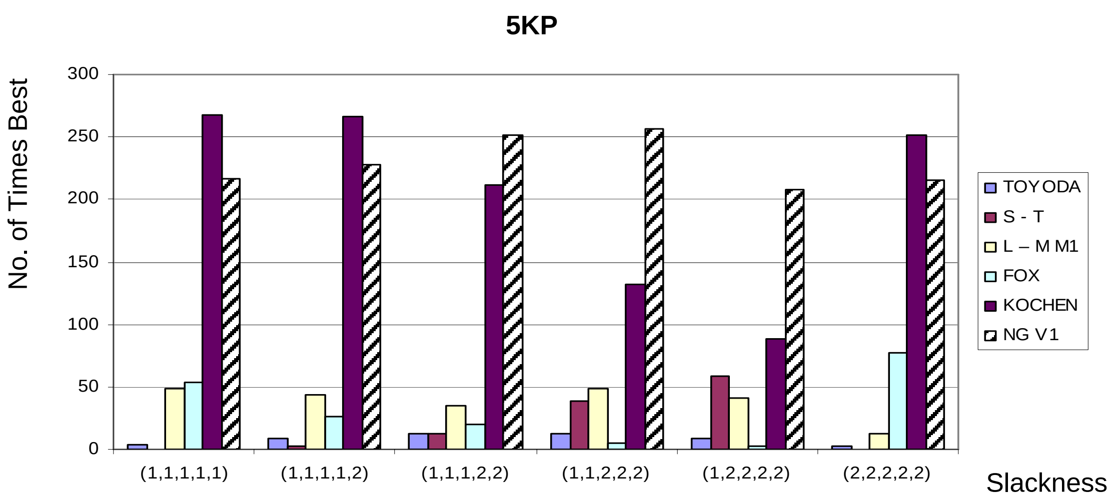


Figure 22. Comparison of NG V1 under Various Constraint Slackness

Table 35. Number of Times Best by NG V1 under Constraint slackness

		NG V1	TOYODA	S – T	L – M M1	FOX	KOCHEN	Sign Test
2KP	(1, 1)	**102**	4	3	7	40	66	*
	(1, 2)	**50**	9	25	8	17	14	*
	(2, 1)	**40**	10	19	3	14	14	*
	(2, 2)	**66**	0	0	10	50	39	*
5KP	(1,1,1,1,1)	**216**	4	0	48	53	268	
	(1,1,1,1,2)	**228**	9	2	43	26	266	
	(1,1,1,2,2)	**252**	12	13	35	20	211	*
	(1,1,2,2,2)	**256**	13	39	49	5	132	*
	(1,2,2,2,2)	**208**	9	58	41	3	89	*
	(2,2,2,2,2)	**215**	3	0	12	77	251	

$\mathbf{H}_0^S$: NG V1 has statistically better performance compared to another heuristic, α =0.1

*: NG V1 is statistically the best among the heuristics compared.

Table 35 and Figure 22 show NG V1 seems to be the best performer for 7 out of 10 constraint slackness settings. The results of NG V1 by various correlation structures also show good performance of NG V1 as compared to the legacy heuristics (see Appendices E.4 and E.5). Counting the number of times a heuristic finds the best solution, excluding ties, by correlation structures, NG V1 heuristic performed better than the previous best heuristic. NG V1 yields the same or better performance in 28 cases out of 45 correlation structures in 2KP and 72 cases out of 126 cases in 5KP. The sign test indicates that NG V1 is statistically the best performer in 16 cases in 2KP and 46 in 5KP. Tables E.4 and E.5 in Appendix E break the performance numbers out by each of the correlation structures in the test problem set.

However, NG V1 does not yield better performance on all combinations between slackness and correlation. Table 35 indicates NG V1 performs worse than KOCHEN 37% of the time. KOCHEN is still the better performer for (1, 1, 1, 1, 1) and (1, 1, 1, 1, 2). FOX also yields better performance in some correlation structures. Thus, the next section presents another heuristic based on KOCHEN that varies constraint weights differently.

5.3.2 New Gradient Heuristic Version 2 (NG V2)

The empirical analysis suggests that a proper constraint weighting scheme affects the solution quality of a heuristic. As a solution process progresses, the better greedy heuristics give more weight to the constraint with the least resources remaining. This allows heuristics to effectively select variables that best use the increasingly limited constraint resources, and ultimately select more variables for inclusion in the solution, and thus improve the objective function value attained. To suitably vary the weight vector, the new gradient heuristic version 2 (NG V2) uses a lognormal point function to compute the weight vector. The *m*-dimensional weight vector (W_{NGV2}) for NG V2 is of the following form:

$$W_{NGV2} = \exp(\sigma \Phi^{-1}(\mathbf{P}_{\mathrm{U}})) \tag{56}$$

where $\sigma = 3$ and Φ^{-1} is the inverse of the normal distribution, vector $\mathbf{P}_{\mathrm{U}}$ is the resource used by the current solution: $\mathbf{P}_{\mathrm{U}} = \left\{ C_i \mid C_i = \sum_{j=1}^{n} x_j a_{ij} \right\}$. The penalty cost function of NG V2 is as follows:

$$v_j = \frac{(A_j \cdot W_{NGV2})}{\left| W_{NGV2} \right|} \tag{57}$$

where the vector A_j provides the resource costs in each constraint for each variable: $A_j = \left\{ a_{ij} \mid i = 1, \dots, m \right\}$. NG V2 selects an item with the largest effective gradient value

$\left(G_j = \dfrac{c_j}{v_j} \right)$ while maintaining feasibility at each iteration. The flow of the solution procedure is identical that of NG V1 as shown in Figure 21.

Before examining raw results, it is informative to examine the constraint weighting trends for NG V2. Figure 23 shows the weighted trend line of NG V2 for the 5KP problems examined. The weight vector of NG V2 responds quite well and quite dynamically. For the slackness setting problems (1, 1, 1, 1, 1), NG V2 weighting mimics both the TOYODA and KOCHEN behavior as seen in Figure 23. For the slackness setting problems (1, 2, 2, 2, 2), NG V2 increases the weight on the dominant constraint earlier in the process than does KOCHEN, but provides the same end-of-the-process emphasis KOCHEN displays in Figure 15.

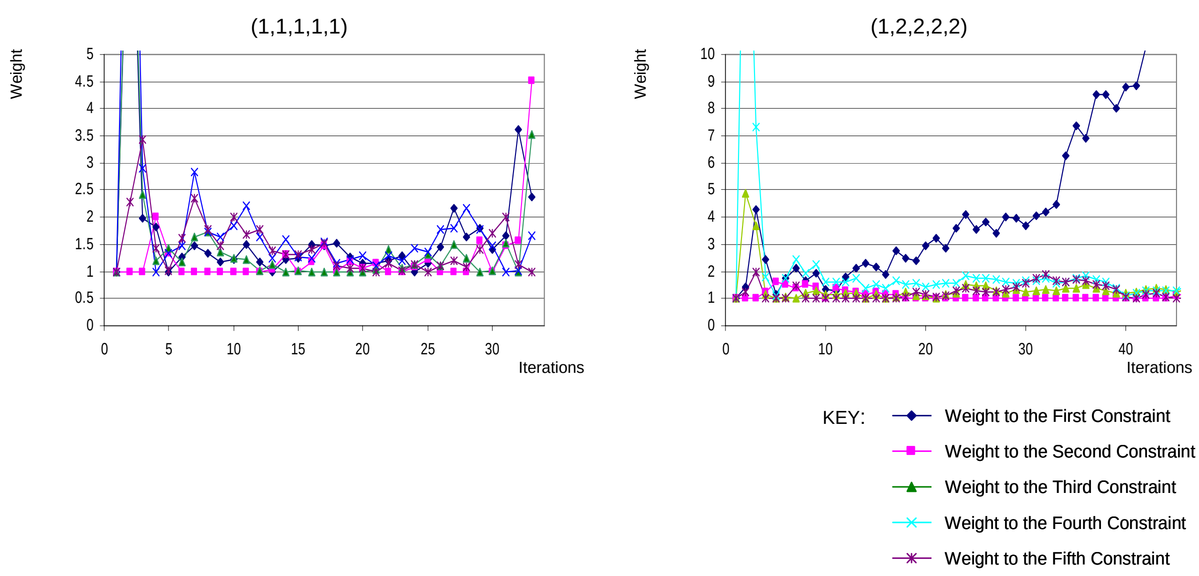


Figure 23. Performance of Weight Trend of NG V2 for (1,1,1,1,1) and (1,2,2,2,2)

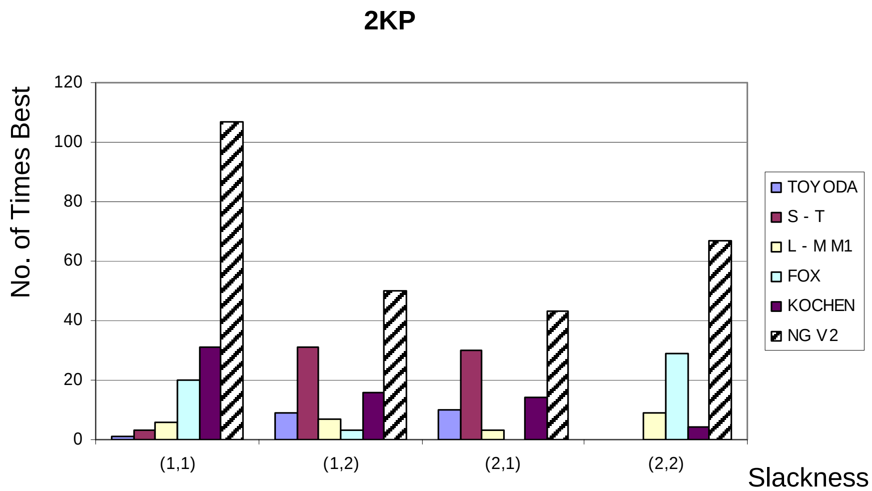
The general comparison of NG V2 heuristic with other legacy heuristics (to include NG V1) is shown in Table 36. The performance of NG V2 heuristic by slackness setting for the 2KP and 5KP test set is summarized in Table 37 and graphed in Figure 24 (ties excluded).

Table 36. Comparison of NG V2 Heuristic with Test Heuristics

Vs Heuristics	Better	Same	Worse
TOYODA	92.4	2.6	5.0
S – T	84.1	5.8	10.1
L – M M1	89.0	2.6	8.5
FOX	97.6	1.2	1.2
KOCHEN	64.7	19.0	16.4
NG V1	55.2	21.4	23.4

(Unit: Percent)

The results show good performance of NG V2 compared to the legacy approaches. Table 36 indicates that NG V2 met or exceeded the performance of the legacy heuristics an average of 91.8 % of the time and that of NG V1 76.6 % of the time. Thus, by varying the weighting scheme, NG V2 appears more effective than NG V1.



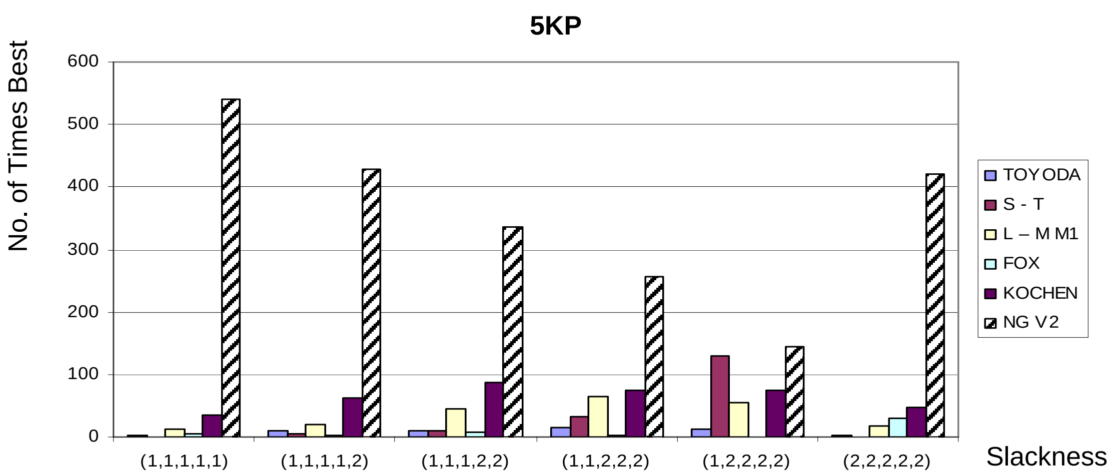


Figure 24. Comparison of NG V2 under Various Constraint Slackness

Table 37. Number of Times Best by NG V2 under Constraint Slackness

		NG V2	TOYODA	S – T	L – M M1	FOX	KOCHEN	Sign Test
2KP	(1, 1)	**107**	1	3	6	20	31	∗
	(1, 2)	**50**	9	31	7	3	16	∗
	(2, 1)	**43**	10	30	3	0	14	∗
	(2, 2)	**67**	0	0	9	29	4	∗
5KP	(1,1,1,1,1)	**540**	2	1	12	4	35	∗
	(1,1,1,1,2)	**428**	9	4	20	3	63	∗
	(1,1,1,2,2)	**335**	10	11	46	7	86	∗
	(1,1,2,2,2)	**256**	16	32	65	3	74	∗
	(1,2,2,2,2)	**145**	12	130	55	0	75	∗
	(2,2,2,2,2)	**421**	2	1	18	31	47	∗

$\mathbf{H}_0^S$: NG V2 has statistically better performance compared to another heuristic, α =0.1

∗: NG V2 is statistically the best among the heuristics compared.

Table 37 and Figure 24 show that NG V2 seems to be the best performer for all constraint slackness levels in the 2KP and 5KP test problems. Ties are not counted in the data presented in Table 37 and Figure 24. Sign tests compare NG V2 with each heuristic, and they indicate that NG V2 yields the best solutions. Table E.6 of Appendix E provides the data for the sign test of NG V2. For the results of correlation structures (See Tables E. 7 and E. 8 in Appendix E), NG V2 yields the same or improved performance as legacy heuristics in 34 cases out of 45 in 2KP and in every 5KP case. Sign test indicates that NG V2 is the best performer in 24 cases (2KP) and in 122 of 126 cases (5KP).

These results suggest that a variable constraint weighting scheme holds promise for devising heuristics that provide improved performance across a range of problem instances. NG V2 exploits a novel constraint weighting mechanism, intended to dynamically respond to constraint resource usage. The results indicate that heuristic effectiveness is a function of properly considering resource utilization during the solution process.

However, Table 37 shows the S – T heuristic still yields comparable performance on mixed constraint slackness settings in 2KP and (1, 2, 2, 2, 2) in 5KP; S – T constantly gives more weight on a tighter constraint starting at the initial iteration, while NG V2 evaluates all constraints to identify a dominant constraint in the early iterations. Thus, the next section introduces NG V3, which uses the merits of preprocessing problem characteristics in order to focus early efforts on a dominant constraint, when such a dominant constraint exists.

5.3.3 New Gradient Heuristic Version 3 (NG V3)

NG V2 improves on the KOCHEN heuristic using the lognormal distribution as a weight vector. Even though NG V2 seems to use a weighting scheme similar to KOCHEN's, NG V2 gives more weight to a dominant constraint in the earlier iterations. Thus, NG V2 outperforms KOCHEN. However, for slackness settings (1, 2), (2, 1) and (1, 2, 2, 2, 2) of Table 37, S – T still produces comparable solution performance compared to NG V2. For one dominant constraint, S – T gives more weight to a dominant constraint during all iterations, while NG V2 gives the same weight to all constraints in the first few iterations, and then increases weight on a dominant constraint. The weighting scheme of NG V2 does not identify a dominant constraint during early iterations.

NG V3 is an improved version of NG V2. NG V3 includes the preprocessing of problem characteristics. The idea of NG V3 is as follows. When problem characteristics are analyzed, a dominant constraint can be identified. Then, the heuristic can pre-weight this constraint to use resources more effectively starting at the initial iteration.

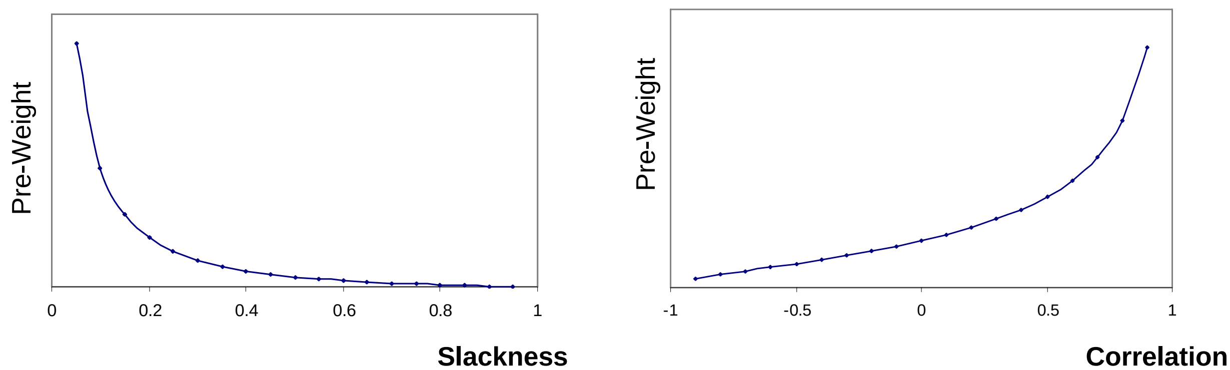


Figure 25. Pre-Weighting Scheme According to Correlation And Slackness

Figure 25 proposes a pre-weighting scheme of NG V3 according to both slackness condition and correlation structure. Based on an analysis of optimal solutions, a dominant constraint (the least resource remaining at the final iteration) can be generated by tight slackness condition and high positive correlation with objective function coefficients. Figure 25 shows that exponential trend lines give a dominant constraint (with tight slackness or a high positive correlation) more pre-weight. The most dominant constraint can have the most pre-weight by multiplying slackness pre-weight and correlation pre-weight. However, pre-weight must balance both slackness and correlation, *i.e.*, a pre-weight for correlation value close to one is almost the same as a pre-weight for tight slackness. Recall that correlations close to one have an influence similar to that of tight constraints. Thus, a high positive correlation value should have a weight similar to that for tight slackness level of a constraint.

Based on empirical testing for selection of pre-weighting parameters, the pre-weighting scheme is as follows:

$$\text{PreWeight}(i) = \exp(\rho_{CA^i}) \times \exp(\frac{1}{r_i}) \tag{58}$$

where r_i is the surplus resource of each constraint $r_i = \sum_{j=1}^{n} a_{ij} - b_i$. Thus, the *m*-dimensional weight vector (W_{NGV3}) for NG V3 is of the following form:

$$W_{NGV3} = \mathbf{P_W} \cdot \exp(\sigma \Phi^{-1}(\mathbf{P_U})) \tag{59}$$

where $\sigma = 3$ (based on computational testing, $\sigma = 3$ provides the best balance between the pre-weight for constraint slackness and pre-weight for correlation structure) and Φ^{-1} is the inverse of the normal distribution, vector $\mathbf{P_U}$ is the resource used by the current solution: $\mathbf{P_U} = \left\{ C_i \mid C_i = \sum_{j=1}^{n} x_j a_{ij} \right\}$, and vector $\mathbf{P_W} = \{ \text{PreWeight}(i) \}$. The penalty cost function of NG V3 is as follows:

$$v_j = \frac{(A_j \cdot W_{NGV3})}{|W_{NGV3}|} \tag{60}$$

where the vector A_j provides the resource costs in each constraint for each variable: $A_j = \{ a_{ij} \mid i = 1, \dots, m \}$. NG V3 selects an item with the largest effective gradient value $\left(G_j = \frac{c_j}{v_j} \right)$ while maintaining feasibility at each iteration. NG V3 solution procedure is shown in Figure 26.

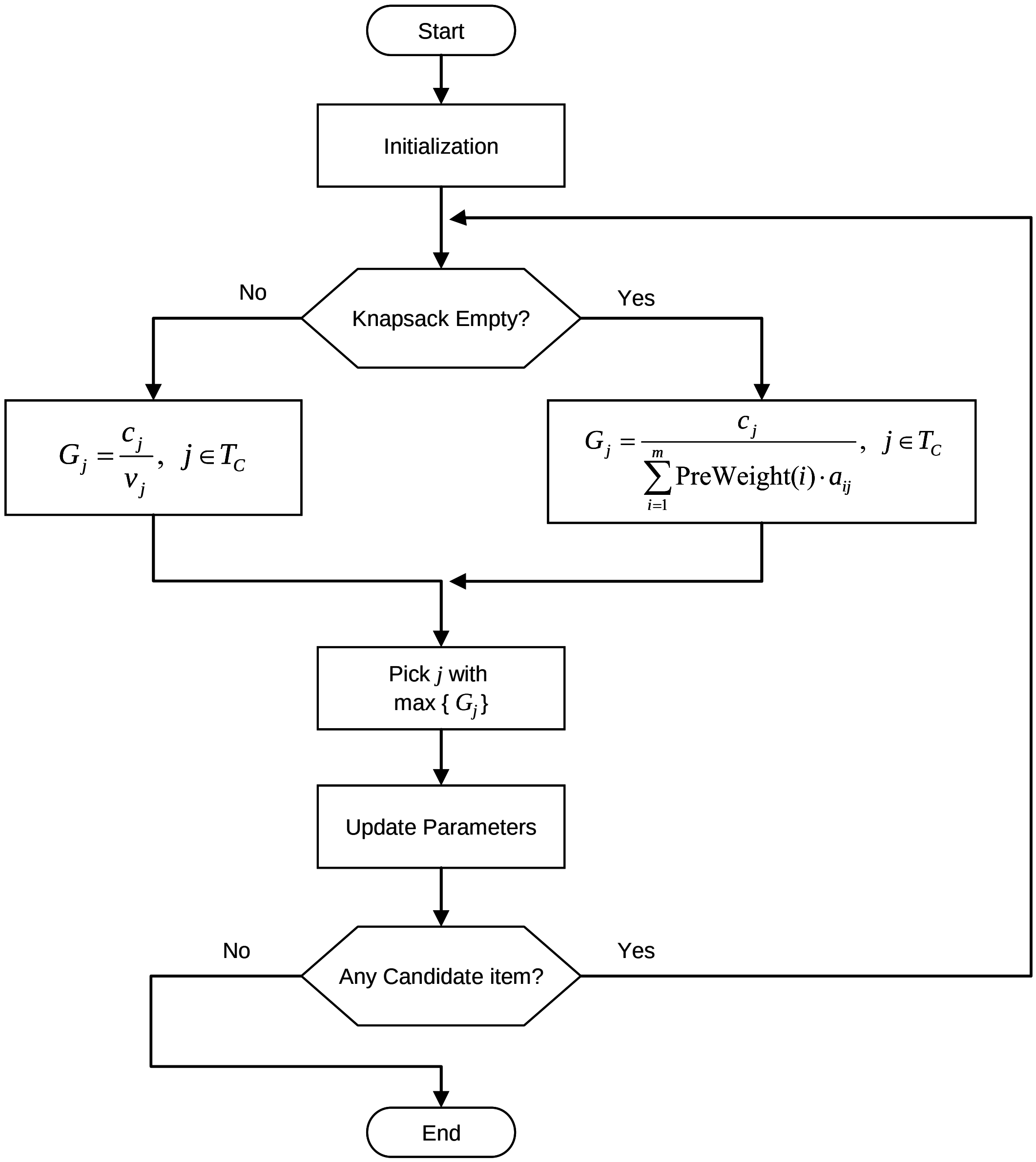


Figure 26. Flowchart of NG V3 Heuristic

The general comparison of NG V3 with legacy heuristics and previous new gradient heuristics on the 2KP and 5KP test set is summarized in Table 38.

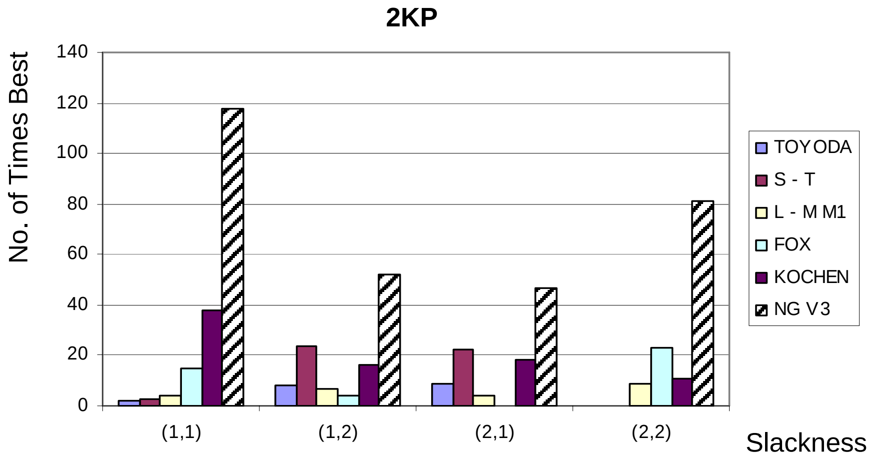
Table 38. Comparison of NG V3 Heuristic with Other Heuristics

Vs Heuristics	Better	Same	Worse
TOYODA	89.1	5.7	5.2
S – T	79.4	11.2	9.3
L – M M1	88.9	4.4	6.7
FOX	90.8	7.3	1.9
KOCHEN	62.6	20.9	16.5
NG V1	58.8	22.9	18.3
NG V2	35.2	48.9	15.9

(Unit: Percent)

The results in Table 38 show the improved performance of NG V3 compared to the legacy approaches, and both new gradient heuristics (NG V1 and NG V2). Even though NG V3 uses the same procedure and equations as NG V2, except multiplying constraints by pre-weights, NG V3 equaled or exceeded the performance of NG V1 81.7% of the time and NG V2 84.1 % of the time. These results suggest that a pre-weighting scheme based on constraint slackness and correlation structure provides improved performance over NG V2 approach.

The overall performance of NG V3 compared to the other heuristics, based on different constraint slackness levels in 2KP and 5KP, is summarized in Figure 27 and Table 39. Ties are not counted in the data presented in Figure 27 and Table 39.



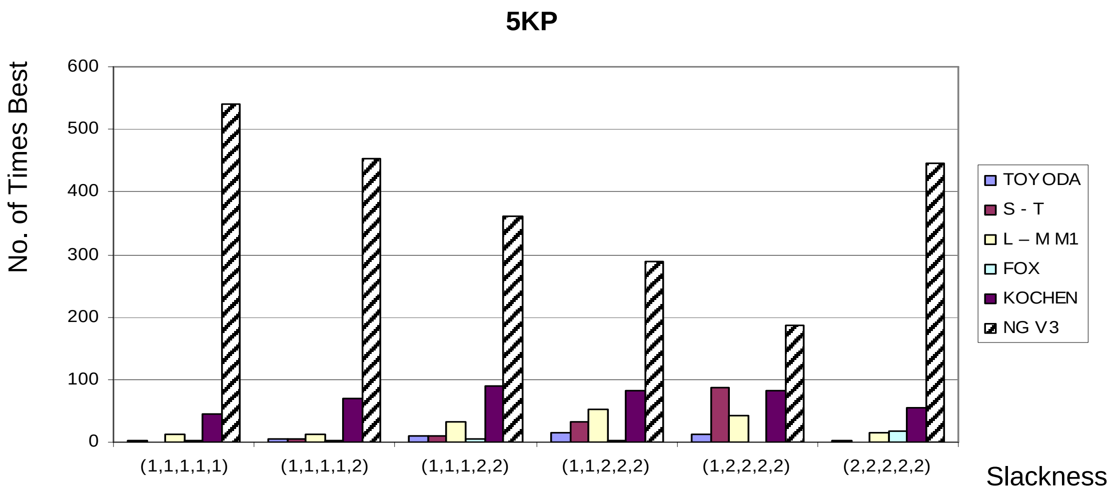


Figure 27. Comparison of NG V3 under Various Constraint Slackness

Table 39. Number of Times best by NG V3 under Constraint Slackness

		NG V3	TOYODA	S – T	L – M M1	FOX	KOCHEN	Sign Test
2KP	(1, 1)	**118**	2	3	4	15	38	∗
	(1, 2)	**52**	8	24	7	4	16	∗
	(2, 1)	**47**	9	22	4	0	18	∗
	(2, 2)	**81**	0	0	9	23	11	∗
5KP	(1,1,1,1,1)	**541**	2	0	12	3	44	∗
	(1,1,1,1,2)	**452**	6	4	13	2	70	∗
	(1,1,1,2,2)	**361**	10	10	32	5	90	∗
	(1,1,2,2,2)	**289**	16	32	52	3	83	∗
	(1,2,2,2,2)	**186**	13	88	42	0	81	∗
	(2,2,2,2,2)	**445**	2	1	14	18	54	∗

$\mathbf{H}_0^S$: NG V3 has statistically better performance compared to another heuristic, α =0.1

∗: NG V3 is statistically the best among the heuristics compared.

For the results of correlation structures (See Tables E. 10 and E. 11 in Appendix E), NG V3 yields the same or better performance with the best legacy heuristic in 39 cases out of 45 in 2KP and 125 cases out of 126 in 5KP. Sign test indicates that NG V3 is the best performer in 22 cases (2KP) and 120 cases (5KP). Table E.9 in Appendix E provides the detailed sign test as shown in Table 39.

The results for constraint slackness and correlations are particularly encouraging. Overall results show NG V3 yields the largest number of best solutions. More importantly, the performance of NG V3 is better than NG V2 on mixed slackness settings in 2KP and (1, 2, 2, 2, 2) in 5KP while maintaining great performance on the various correlation structures. The goal of developing NG V3 is to improve performance of NG V2 when one constraint is made dominant by slackness setting.

5.4 New Reduction Heuristic

Based on the empirical analysis of legacy transformation heuristics, Chapter IV showed that local improvement and transforming MKP to KP using dual variables are a very useful and effective way to solve MKP. The Lagrangian relaxation of MKP with an m – dimensional vector of Lagrange multipliers $\boldsymbol{\mu}$ is defined as follows:

$$Z_{LR}(\boldsymbol{\mu}) = \max\left\{\sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{m} \mu_i \left(b_i - \sum_{j=1}^{n} a_{ij} x_j \right)\right\} \tag{61}$$

where $x_j = 0$ or 1, $\mu_i \geq 0$, $i = 1, 2, ..., m$, $j = 1, 2, ..., n$.

The solution to problem (61) provides an upper bound on the optimal solution of the original MKP shown in Equations (2) through (4). The best bound, *i.e.,* the upper bound that is the closest to the optimal solution of the original MKP, is determined by finding multipliers $\boldsymbol{\mu}$ which correspond to

$$Z_{LD}(\boldsymbol{\mu}^*) = \min_{\mu} Z_{LR}(\boldsymbol{\mu}) \ . \tag{62}$$

This problem is called the Lagrangian dual problem. If the optimal multipliers $\boldsymbol{\mu}^*$ are found, the optimal solution of (62) is an upper bound on the optimal solution of the original MKP shown in Equations (2) through (4). This upper bound with optimal multipliers $\boldsymbol{\mu}^*$ is tighter than the bound provided by the LP relaxation of the MKP (Nemhauser and Wolsey , 1988). However, finding optimal multipliers $\boldsymbol{\mu}^*$ is a difficult combinatorial problem. Dual variables from the solution of the linear programming relaxation of the original MKP can be used as good multipliers (Fisher, 1981); this research uses these dual variables as the multipliers.

Once the multipliers have been found, the Lagrangian relaxation problem, Equation (61), can be solved directly. The coefficient of the *j*th variable is as follows:

$$\gamma_j = c_j - \sum_{i=1}^{m} \mu_i a_{ij} \tag{63}$$

If γ_j is positive, the *j*th variable increases the objective function value. If γ_j is negative, the *j*th variable decreases the objective function value. Therefore, set x_j to 1 if its coefficient is positive and x_j to 0 if its coefficient is non-positive. This solution is the optimal solution of $\boldsymbol{Z_{LR}}(\mu)$.

In developing heuristic algorithms for the KP, Balas and Zemel (1980) found that the solution obtained by a greedy heuristic using the decreasing order of bang-for-buck ratios (benefit/cost) differed from the optimal solution among only a few variables. Comparing optimal and heuristic solutions, some variables are always set to zero and some variables are always set to one. There is, however, a subset of variables where the approaches differ and the bang-for-the-buck greedy heuristic has insufficient discriminatory power. Psinger (1999) called this subset the core problem. Pirkul (1987) and Psinger (1999) focus on core problem issues. Pirkul defined a core problem for MKP as "the subproblem in those variables whose $\frac{c}{\mu A}$ ratio (*c* is the objective function coefficient vector, *A* is the constraint coefficient vector, and μ is a positive multiplier vector) falls between the maximum and minimum ratios for which $\boldsymbol{x}$ has a different value in an optimal solution to MKP from that in an optimal solution to linear programming relaxation." Pirkul also suggested that the core problem consists of very few variables compared to the original problem. Thus, very few variables in the solution of the Lagrangian problem, Equation (61), are different from those in an optimal solution to the original MKP.

Using the solution of the Lagrangian problem, Equation (61), variables can be categorized as selected, non-selected, or uncertain. Since dual variables include information regarding constraint slackness and correlation structures, γ_j of the Lagrangian problem, Equation (63), function like the effective gradient value of the greedy heuristics. If γ_j of Equation (63) has a relatively large positive value, the optimal solution to the original MKP has a tendency to have $x_j = 1$ while for a relatively large negative value, $x_j = 0$. However, if γ_j is near zero, the x_j value in the optimal solution is uncertain. The core problem consists of these uncertain variables. Once identified, the core problem can be solved using a heuristic or exact algorithm. A new reduction approach procedure can be summarized as follows:

Step 1: Solve the linear programming relaxation to find Lagrange multipliers

Step 2: Transform a MKP into an unconstrained problem (only binary variable restriction) using a Lagrangian relaxation

Step 3: Categorize as selected, unselected, and uncertain variables according to γ_j from the Lagrangian relaxation problem

Step 4: Solve the core problem using a heuristic or exact algorithm

Balas and Zemel (1980) suggest, "for the KP the size of the core problem was a small fraction of the full problem size and almost independent of the problem size". Pirkul (1987) presented the number of variables in the core problem (number of variables have different values in an optimal solution to MKP) as shown Table 40. Table 40 suggested that the core problem consists of few variables with respect to the original problem and the core problem size increases at a much slower rate than the increase in original problem size.

Table 40. No. of Variables in Core Problem for 3KP (Pirkul, 1987)

n	No. of variables in Core Problem								
	Slackness S_i =0.5			Slackness S_i =0.37			Slackness S_i = 0.25		
	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
100	1	7.6	11	5	9.1	14	3	9.1	15
200	4	11.2	19	6	14.0	22	5	12.4	21
300	5	15.4	26	6	14.6	25	10	15.7	30
400	8	15.0	28	4	14.8	32	8	17.3	27
500	10	17.1	27	1	16.3	33	16	18.7	34
600	16	18.6	23	6	15.2	30	17	21.3	31

(Pirkul, 1987)

The empirical testing indicates that the solution obtained by a simple greedy heuristic using the decreasing order of Equation (63), $\gamma_j = c_j - \sum_{i=1}^{m} \mu_i a_{ij}$, differed from the optimal solution by only 0 to 11 variables (Average 3.15) in 2KP and 0 to 14 variables (Average 4.56) in 5KP sets. Fewer variables differ from the optimal solution in our results than in Pirkul's results because our test problems include some problems for which a heuristic obtains the optimal solution.

While analyzing the core problem, Pirkul's results (1987) show that 15% of the number of variables seem reasonable to use as the core problem. This was reasonable given his limited test problem set. However, a more robust problem set will yield instances where a possibly larger percentage of variable have G_j clustered around 0. Thus, a more reasonable core problem definition involves some percentage of the effective gradient range. For example, hard test problems (*e.g.*, when ρ = (2, 2, 2)) should involve more variables in the core problem while easy test problems (*e.g.*, ρ = (-2, -2, 2)) should involve fewer variables. Examining problems using Equation (63) demonstrates this property in that most values of γ_j are far from zero for easy test problems (*i.e.,* obvious discriminatory power), while, for hard test problems, more values of Equation (63) are located around zero. Figure 28 shows the density of γ_j values (*i.e.,* each value of γ_j, $j = 1, \ldots, n$) according to a representative hard problem $\rho = (2, 2, 2)$ and easy problem $\rho = (-2, -2, 2)$ from the 2KP. All values of γ_j are scaled and based on a maximum value of 1 and a minimum value of -1.

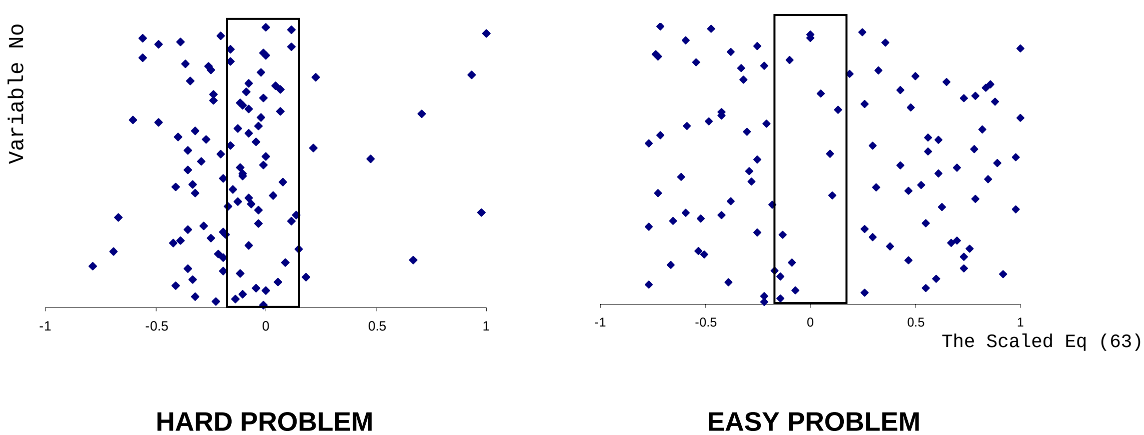


Figure 28. Density of Variables around Zero Value by Hard and Easy Problem

Figure 28 implies hard problems have more variables near zero than do easy problems. In the example of Figure 28, the core problem of the hard problem includes 45 variables in the range of [-0.15, 0.15], while the core problem of the easy problem includes 12 variables. Thus, the core problem should vary according to problem difficulty (constraint slackness and correlation structures). The range of [-0.15, 0.15] of γ_j is selected for the core problem, where the variables have the value of $\gamma_j = c_j - \sum_{i=1}^{m} \mu_i a_{ij}$ falling in this range. Considering 15% of the value range for γ_j is more generous and effective than considering just 15% of the variables as part of the core problem because the number of variables in the core problem varies according to various problem characteristics. When the core problem contains the variables, x_j, such that $\gamma_j \in [-0.15, 0.15]$, overall core problem consists of an average of 27.5 variables in the 2KP case and 28.1 variables in the 5KP case. The range of [-0.15, 0.15] includes all variables having different values from optimal solutions in 93 % of 2KP problems (1042 of 1120 problems) and 96% of 5KP problems (3625 of 3780 problems). The range of γ_j can be expanded to include more variables; however, this increase makes the larger core problem more difficult to solve. In the 2KP, especially, extremely high positive correlation structures ($\rho_{CA} \approx 0.99$) cause outliers, which have a value of zero in the optimal solution while having a higher positive value of γ_j. However, these are extreme correlation structures, so the range of [-0.15, 0.15] is considered sufficient as it contains 96.7 % of the proper variables in the 2KP problems.

Solving the core problem has the advantage of fewer variables, so branch-and-bound is usually applied. A greedy approach is used in this research. Since all variables in the core problem have nearly equal values of $\gamma_j = c_j - \sum_{i=1}^{m} \mu_i a_{ij}$, a greedy heuristic will have less discriminatory power for selecting correct variables. Since NG V3 is the best heuristic among the greedy heuristics (having the most discriminatory power), NG

V3 is used to solve the core problem. Table 41 shows the overall comparison of this new reduction heuristic (NR) based on different constraint slackness levels in 2KP and 5KP. The details for sign test are provided at Table E.12 in Appendix E and the detailed results for various correlation structures by the new reduction heuristic (NR) are shown at Tables E.13 and E.14 in Appendix E. Ties are not counted in Table 41.

Table 41. Comparisons to New Reduction Heuristic (Core Solved by NG V3)

		NR (NGV3)	TOYODA	S – T	L – M M1	FOX	KOCHEN	Sign Test
2KP	(1, 1)	**110**	3	3	5	17	58	∗
	(1, 2)	**50**	8	23	6	8	19	∗
	(2, 1)	**48**	9	21	4	2	17	∗
	(2, 2)	**72**	0	0	8	13	31	∗
5KP	(1,1,1,1,1)	**538**	2	1	11	2	54	∗
	(1,1,1,1,2)	**437**	5	4	17	1	120	∗
	(1,1,1,2,2)	**352**	8	9	38	4	144	∗
	(1,1,2,2,2)	**270**	18	31	49	3	131	∗
	(1,2,2,2,2)	**206**	12	71	37	0	97	∗
	(2,2,2,2,2)	**451**	1	0	11	9	72	∗

$\mathbf{H}_0^S$: NR (NG V3) has statistically better performance compared to another heuristic, α=0.1
∗: NR (NG V3) is statistically the best among the heuristics compared.

Table 41 suggests that the new reduction heuristic (NR) based on the core problem concept is robust against various constraint slackness and correlation structures as compared to the set of legacy heuristics.

5.5 Reduction Heuristic with an Improvement Phase

While the new reduction approach is effective, it is a single path algorithm like other greedy heuristics. Based on the empirical analysis of Chapter IV, Pirkul's (1987) improvement phase is more efficient than Glover's (1977), so it is added to NG V3 after

NG V3 provides a feasible solution for the core problem. Figure 29 shows the flowchart of the new reduction heuristic with Pirkul's improvement phase (NR(P)).

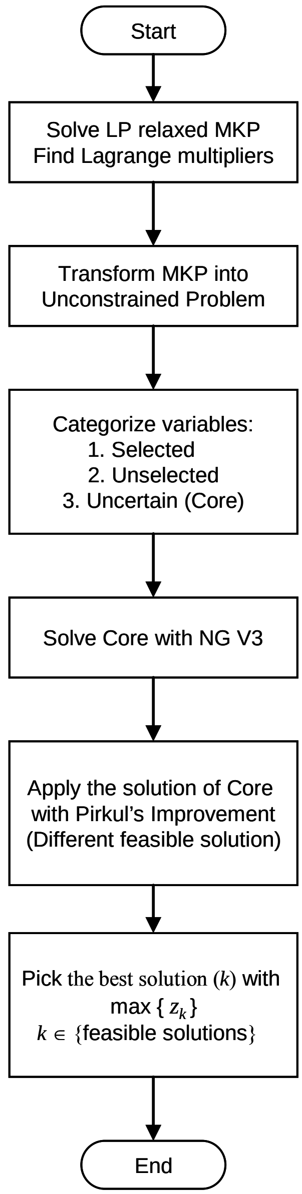


Figure 29. Flowchart of NR (P) Heuristic

Table 42 shows the solution performance of this new heuristic, NR(P), with its improvement phase compared to each greedy heuristic.

Table 42. Number of Times Best by New Reduction Heuristic (with improvement) under Constraint Slackness

		NR (P)	TOYODA	S – T	L – M M1	FOX	KOCHEN
2KP	(1, 1)	**217**	0	0	1	1	7
	(1, 2)	**156**	4	4	1	1	5
	(2, 1)	**146**	2	6	0	0	4
	(2, 2)	**167**	0	0	4	3	6
5KP	(1,1,1,1,1)	**604**	0	0	2	0	8
	(1,1,1,1,2)	**571**	1	2	4	0	18
	(1,1,1,2,2)	**518**	4	3	4	3	31
	(1,1,2,2,2)	**506**	2	6	13	1	22
	(1,2,2,2,2)	**444**	3	13	3	0	17
	(2,2,2,2,2)	**534**	1	0	2	1	18

Key:
NR(P) New reduction heuristic with Pirkul's improvement phase

Table 42 indicates that NR(P) is more robust over all constraint slackness levels than the legacy heuristics. The size of the core problem is small, so comparison of new feasible solutions is also small. Thus, a conclusion is that the core problem reduction is an effective technique, saving computation time and improving solution quality. Table 43 shows a comparison of the new reduction heuristics with Glover's and Pirkul's heuristics. In Table 43, the column of average numbers of comparisons shows average numbers of feasible solutions created by the heuristics in order to select the best solution among them.

Table 43. Comparison of New Reduction Heuristics with Test Heuristics

Heuristics	Average No. of Comparisons	Average Solution Quality
New Reduction (NR)	1	99.52 % of Optimum
NR w/ Pirkul's Improvement Phase (NR(P))	13.4	99.88 % of Optimum
Pirkul Heuristic (PIRKUL)	47.0	99.89 % of Optimum
Glover Heuristic (GLOVER)	50	99.80 % of Optimum
Best Legacy	1	99.05 % of Optimum

Table 43 shows that the new reduction heuristic with Pirkul's improvement (NR(P)) has 28.5 % of PIRKUL's comparisons (13.4 vs. 47.0 comparisons), but produces similar solution quality (0.01 % difference). Compared to Glover's heuristic, NR(P) yields better solutions (0.08 % difference) with 26.8% of GLOVER's comparisons (13.4 vs. 50 comparisons). The core problem concept yields similar solution quality with fewer comparisons than Pirkul's and Glover's heuristics. More importantly, this core problem can be solved by any type of algorithm such as a meta-heuristic or exact algorithm.

5.6 Comparison of All New Heuristics

Six different types of heuristics based on the empirical analysis of legacy greedy heuristics have been discussed: TYPE, NG V1, NG V2, NG V3, NR, NR(P). Table 44 shows the comparison of the newly developed heuristics. Since these heuristics yield solutions close to optimal, the comparisons in Table 44 are based on the average relative errors (smaller values are better).

Table 44. Comparison of All New Heuristics by Average Relative Error

	Slackness	TYPE	NG V1	NG V2	NG V3	NR	NR (P)
2 KP	(1, 1)	0.784	0.967	0.479	0.455	0.599	0.116
	(1, 2)	0.278	0.209	0.220	0.197	0.213	0.039
	(2, 1)	0.203	0.174	0.187	0.153	0.156	0.026
	(2, 2)	0.183	0.230	0.105	0.091	0.118	0.021
5 KP	(1,1,1,1,1)	3.741	3.252	1.036	0.731	0.809	0.308
	(1,1,1,1,2)	1.932	2.017	0.827	0.659	0.749	0.230
	(1,1,1,2,2)	1.178	1.131	0.621	0.512	0.630	0.183
	(1,1,2,2,2)	0.725	0.617	0.475	0.415	0.513	0.115
	(1,2,2,2,2)	0.925	0.312	0.484	0.359	0.350	0.076
	(2,2,2,2,2)	0.736	0.631	0.258	0.184	0.162	0.054
Total Average		**1.270**	**1.114**	**0.533**	**0.419**	**0.475**	**0.117**
% of Optimum		98.73 %	98.89 %	99.47 %	99.58 %	99.52 %	99.88 %

Key:	
TYPE	A Typed Heuristic
NG V1	New Gradient Heuristic Ver.1
NG V2	New Gradient Heuristic Ver.2
NG V3	New Gradient Heuristic Ver.3
NR	New Reduction Heuristic (No Improvement Phase)
NR(P)	New Reduction Heuristic with Pirkul's Improvement Phase(1987)

The TYPE heuristic uses legacy heuristics only and yields the worst performance among the new heuristics. Among new greedy approaches, NG V3 demonstrates the best performance. The overall best performer is NR(P), problem reduction with local improvement.

VI. Computational Test on Randomly Generated Problem Sets

6.1 Introduction

In previous chapters, evidence was presented that problem characteristics, constraint slackness and correlation structures affect the solution performance of heuristics. Since the commonly used existing standard benchmark MKP set does not include sufficiently diverse problem characteristics, experimental information regarding the solution performance of heuristics is restricted. Thus, this chapter provides a new MKP test set that varied all desired problem characteristics for a structured empirical test of heuristic solution performance. Legacy heuristics and new heuristics are used to solve the problems in these new MKP test sets. This information is used to draw final conclusions.

6.2 Test Problem Generation

Most published empirical studies compare heuristic performance using test problems. Legacy standard problems are available as are de facto standard problem sets. Most are available via the internet. Many researchers develop randomly generated problem sets as a part of their research to verify their algorithm. Few researchers have actually systematically studied the effects of problem characteristics among the test problems. To promote potential use in the research community and to objectively evaluate heuristic performance on a new test set, the test set generated follows the general structure of Chu and Beasley (1998) problem set, the set available at Beasley's (2004) web site.

In the new MKP test set, five problem generation parameters are varied: number of variables, number of constraints, the constraint slackness, the correlation value, and coefficient distribution. Nine test files containing 270 problems are created (30 problems in each file). Each file has a different combination of number of variables and number of constraints as follows: 50-5KP, 100-5KP, 250-5KP, 50-10KP, 100-10KP, 250-10KP, 50-25KP, 100-25KP, and 250-25KP (Number of Variables-Number of Constraints). Table 45 provides the general information regarding the new MKP test set.

Table 45. General Information of New MKP Test

Parameters	Values
No. of Problems	270 problems (9 files, 30 problems each)
No. of Variables	50, 100 , 250
No. of Constraints	5, 10, 25
Slackness	Randomly generate slackness ratio, s_i ~ Unif (0.2, 0.8), for ith constraint
Correlation	Randomly generate correlation value, ρ_{CA^i} ~ Unif (-0.9, 0.9) between objective function coefficient and ith constraint coefficient. Set each $\rho_{A^iA^j}$ as midpoint of range defined by ρ_{CA^i} and ρ_{CA^j}
Coefficient Distribution	Objective function coefficient c_j for all j ~ Discrete Unif (1, 100) Each constraint coefficient a_{ij} for all i, j ~ Discrete Unif (1, r_i) where r_i ~ Discrete Unif (40, 90)

Remark:
Unif: Uniform Distribution
Discrete Unif: Discrete Uniform Distribution

As shown in Table 45, slackness ratios and correlation values are randomly generated. Thus, slackness and correlation can truly cover the range from their minimum value to maximum value. The correlation of inter-constraint, $\rho_{A^iA^j}$, representing the correlation between *i*th constraint coefficient and *j*th constraint coefficient, is fixed by Equation (42), $\rho_{A^iA^j} = \rho_{CA^i} \cdot \rho_{CA^j}$, to maintain the correlation matrix $\mathfrak{R}$ for test problem coefficients as positive semidefinite by Theorem 4. Since objective function coefficients and each constraint's coefficient follows different distributions, each MKP instance is different.

A generation scheme to create the new MKP test set follows the Procedure CorrGeneration and the Procedure Iman and Conover approach used to generate the 5KP test problem sets used in Chapter III. These procedures generated all required problem characteristics for the 5KP test set. The procedure for generating the new test problem set is:

Procedure New Test Problem Generation

1. Randomly generate r_i for all i ~ Discrete Unif (40, 90).
2. Randomly generate ρ_{CA^i} for all i ~ Unif (-0.9, 0.9).
3. Generate the correlation matrix $\mathfrak{R}$ using Procedure CorrGeneration.
4. Using $\mathfrak{R}$ and the r_i, generate objective function coefficients, c_j for all j ~ Discrete Unif (1, 100), and constraint coefficients, a_{ij} for all i, j ~ Discrete Unif (1, r_i).
5. Randomly generate slackness ratio, S_i for all i ~ Unif (0.2, 0.8).
6. Set each RHS value, $b_i = S_i \sum_{j=1}^{n} a_{ij}$ for all i.

6.3 New Test Set vs. Legacy Test Set

This section introduces characteristics of the new test set by comparing two existing legacy test sets. Legacy generation approaches and standard test problem sets do not include various constraint slackness and correlation structures among the problem coefficients.

Martello and Toth (1988, 1997) devised three classes of correlation to check computational performance for knapsack problems:

Uncorrelated: w_j uniformly random in [1, a],

p_j uniformly random in [1, a],

Weakly Correlated: w_j uniformly random in [1, a],

p_j uniformly random in [w_j - δ, w_j + δ]

Strongly Correlated: w_j uniformly random in [1, a],

$p_j = w_j + \delta$

where p_j = profit of item j, w_j = weight of item j given n items, and a and δ are prefixed constants. These generation schemes are simply linear functions and are labeled implicit correlation induction strategies (Cario *et al*., 2002). Although these have been widely used in empirical studies (for many types of optimization problems), most researchers never quantify the level of correlation induced between the problem coefficients during the studies. The weakly correlated scheme produces correlation values with $\rho = 0.98$, and the strongly correlated scheme produces correlation coefficients with $\rho = 0.99$ (Cario, *et al*., 2002).

The second example is the Beasley (2004) problem set presented in Chapter III and first developed for Chu and Beasley (1998). Chu and Beasley (1998) generated the problem coefficients in the following manner. The constraint coefficients were randomly generated *Unif* (0,1000). The objective function coefficients were then set using the formula:

$$c_j = \frac{\sum_{i=1}^{m} a_{ij}}{m + 500q_j} \tag{64}$$

where the q_j is uniformly distributed *Unif* (0,1), essentially noise. The intent is to induce correlation between the objective function and constraint coefficients, but these problems do not adequately vary the correlation coefficient between objective function and *i*th constraint coefficient, ρ_{CA^i}. This approach implicitly induces a correlation level of approximately 0.42, and sampling error provides a narrow range about this value. Figure 5 showed the correlation range between objective function coefficients and each of the five constraint coefficients in one of the Beasley test problem files. The narrow range of ρ_{CA^i} can restrict the extensibility of solution performance of a heuristic beyond that observed with the test problem set (even though extensibility is quite often implied in the discussions). This is particularly worrisome given that the first example of problem generation restricts the correlation to the 0.98 to 0.99+ range.

On the contrary, the new test set covers the correlation range from -0.9 to 0.9. Table 46 and Figure 30 compare the correlation range of the new test set and Beasley's (2004) test set.

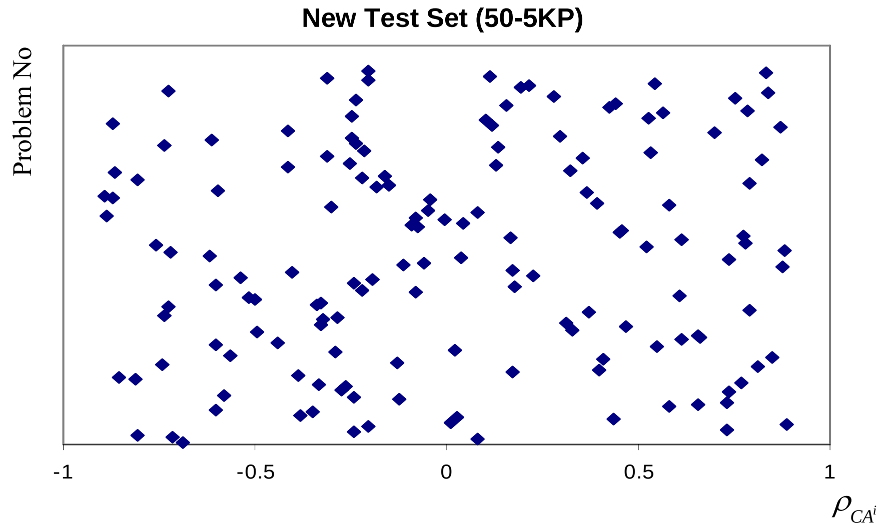
Table 46 shows the minimum of ρ_{CA^i} and the maximum of ρ_{CA^i} in each test set (Beasley's data is the same as that of Table 1).

Table 46. Correlation Analysis of Standard Problems

New Test Set				Beasley's Test Set			
File	min ρ_{CA^i}	max ρ_{CA^i}	(n,m)	File	min ρ_{CA^i}	max ρ_{CA^i}	(n,m)
50-5KP	-0.895	0.885	(50, 5)	*mknapcb1*	0.094	0.511	(100,5)
100-5KP	-0.883	0.892	(100, 5)	*mknapcb2*	0.163	0.461	(250,5)
250-5KP	-0.875	0.871	(250, 5)	*mknapcb3*	0.189	0.403	(500,5)
50-10KP	-0.884	0.899	(50, 10)	*mknapcb4*	-0.157	0.459	(100,10)
100-10KP	-0.881	0.882	(100, 10)	*mknapcb5*	0.003	0.326	(250,10)
250-10KP	-0.887	0.883	(250, 10)	*mknapcb6*	0.030	0.308	(500,10)
50-25KP	-0.898	0.905	(50, 25)	*mknapcb7*	-0.256	0.437	(100,30)
100-25KP	-0.902	0.904	(100, 25)	*mknapcb8*	-0.192	0.307	(250,30)
250-25KP	-0.889	0.890	(250, 25)	*mknabcb9*	-0.074	0.213	(500,30)

(n,m) represents (variables, constraints) in problems

Figure 30 graphically shows the correlation range of the 50-5KP file (new test set) and *mknapcb1* (Beasley's test set). Both files include 30 problems. Since all problems are 5KP, each problem has five correlation points in Figure 30; each graph has 150 points.



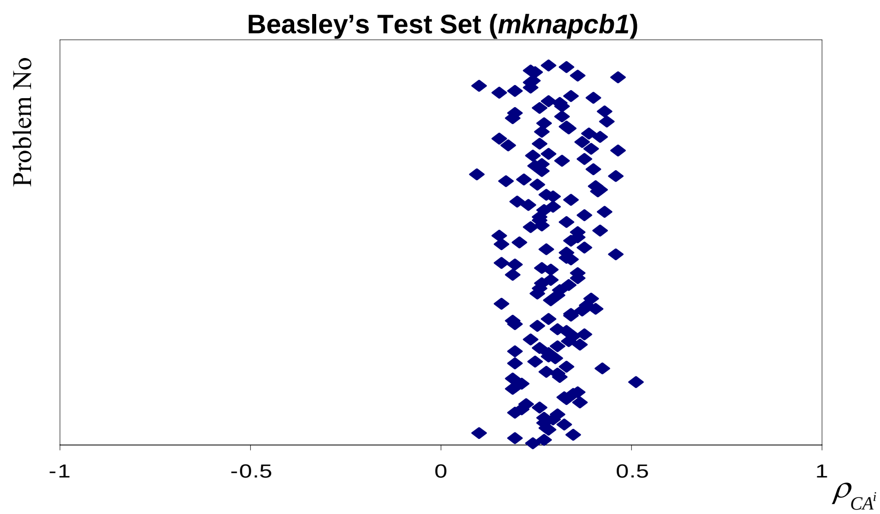


Figure 30. Correlation Range Comparison of 50-5KP and *mknapcb 1*

Figure 30 indicates that the correlation range of the new problem is very wide, while the correlation range of Beasley's test set is very narrow. An adequate test set should provide a full range of test instances.

Figure 31 shows that the slackness range of the new test problem set spreads out from 0.2 to 0.8, while problems from the Beasley set do not.

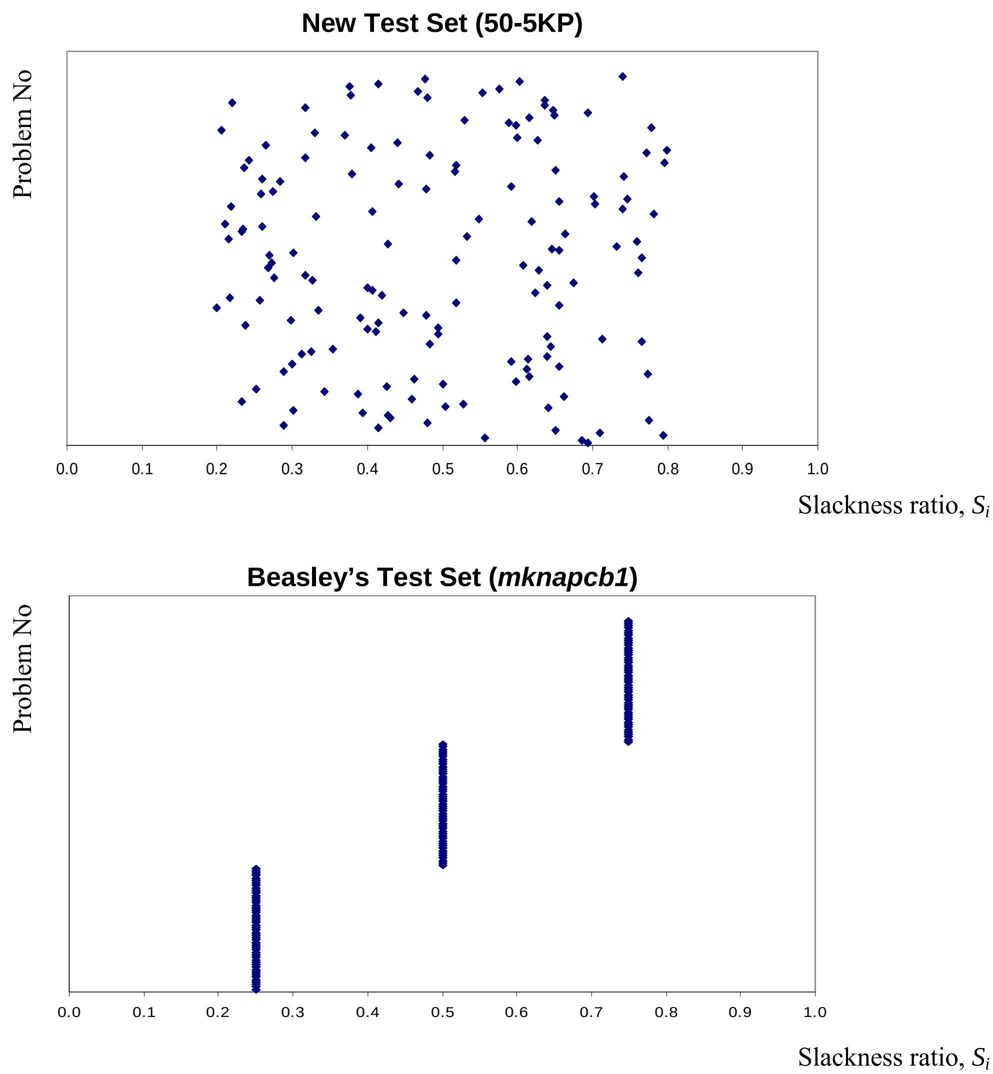


Figure 31. Slackness Range Comparison of 50-5KP and *mknapcb 1*

For slackness ratios, Beasley's test set uses the convention of equal slackness ratios applied to all constraints although varied across three ratios (0.25, 0.5, 0.75) within a problem set file. Thus, all constraint slackness ratios are equal in each problem with different slack ratios (0.25, 0.5, 0.75) applied to 10 problems in each 30 problem test set.

Existing benchmark test sets do not provide a sufficiently diverse set of test problems. This means testing may not gain full experimental information regarding heuristic solution procedure performance. Therefore, the new test set is created to provide a more comprehensive test set and a true range of representative MKPs.

6.4 Computational Results

The overall performance of five legacy heuristics and all new heuristics on Beasley's test set and on the new test set is summarized in Tables 47 and 48, respectively. Both Tables present average relative errors which is the percentage from the optimal solution for each heuristic solution. Optimal solutions in the new test set were achieved using Xpress commercial software. Beasley's test set provides the optimal solution of each problem.

Table 47. Average Relative Error by Each Heuristic Solving Beasley's Problem Set (2004)

Test Problem	TOYODA	S – T	L – MM1	FOX	KOCHEN	TYPE	NGV1	NGV2	NGV3	NR	NR(P)
mknapcb1 (100-5KP)	2.811	3.557	5.736	9.469	0.975	0.975	1.444	1.087	0.991	1.172	0.532
mknapcb2 (250-5KP)	2.095	2.456	6.576	9.823	0.440	0.440	0.977	0.396	0.406	0.579	0.242
mknapcb3 (500-5KP)	1.474	1.989	6.927	9.873	0.207	0.207	0.866	0.201	0.218	0.265	0.078
mknapcb4 (100-10KP)	3.889	4.463	4.868	11.492	1.809	1.809	2.106	1.522	1.634	1.505	1.103
mknapcb5 (250-10KP)	2.713	3.383	4.355	11.259	0.809	0.809	1.096	0.663	0.605	0.668	0.484
mknapcb6 (500-10KP)	1.908	2.242	4.426	10.902	0.316	0.316	0.683	0.306	0.275	0.307	0.190
mknapcb7 (100-30KP)	4.874	5.359	3.199	12.717	2.250	2.250	2.782	1.684	1.730	1.455	1.452
mknapcb8 (250-30KP)	3.744	3.900	2.867	12.763	1.390	1.390	1.708	0.986	1.005	0.864	0.801
mknapcb9 (500-30KP)	2.822	2.961	2.560	12.057	0.793	0.793	1.115	0.653	0.587	0.511	0.491
Total Average	2.926	3.368	4.613	11.151	0.999	0.999	1.420	0.833	0.828	0.814	0.597

(Unit: Percent)

NG : New Gradient Heuristic
NR: New Reduction Heuristic
NR(P): New Reduction Heuristic with Pirkul's Improvement

Table 48. Average Relative Error by Each Heuristic Solving New Test Set

Test Problem	TOYODA	S – T	L – MM1	FOX	KOCHEN	TYPE	NGV1	NGV2	NGV3	NR	NR(P)
50-5KP	4.275	5.820	2.431	7.744	2.221	2.953	1.635	1.101	0.834	1.001	0.219
100-5KP	5.508	6.422	3.937	6.155	1.772	2.083	0.943	0.450	0.403	0.472	0.097
250-5KP	5.844	5.921	6.177	5.475	0.889	1.065	0.649	0.267	0.210	0.196	0.038
50-10KP	6.550	6.758	3.210	12.049	3.398	3.398	3.729	1.806	1.685	2.326	0.921
100-10KP	7.574	7.688	3.938	10.486	2.662	2.662	1.942	0.940	0.832	0.939	0.267
250-10KP	10.461	10.673	7.664	10.179	2.451	2.441	1.595	0.629	0.344	0.318	0.075
50-25KP	9.148	10.546	4.457	15.322	6.840	7.408	5.442	1.910	1.827	2.238	0.930
100-25KP	10.518	12.848	5.645	13.991	5.754	6.033	5.095	1.504	1.139	1.082	0.541
250-25KP	13.242	14.722	8.514	13.744	5.022	5.022	4.348	1.019	0.507	0.517	0.176
Total Average	8.124	9.044	5.108	10.572	3.445	3.674	2.820	1.070	0.865	1.010	0.363

(Unit: Percent)

NG : New Gradient Heuristic
NR: New Reduction Heuristic
NR(P): New Reduction Heuristic with Pirkul's Improvement

Tables 47 and 48 present empirical results supporting our three important premises: (1) the new test set provides a true range of problem characteristics leading to relatively robust conclusions, (2) Beasley's benchmark set of 270 problems is not fully adequate, so heuristic performance conclusions are limited, and (3) the new heuristics yield robust solution quality over a complete range of problem characteristics.

Table 47 results indicate KOCHEN is a best choice among the legacy heuristics examined. Moreover, KOCHEN behavior, in terms of solution quality, is fairly consistent as the problem size increases. The TYPE heuristic matches KOCHEN performance which is not unexpected given the range of problems in the test set. Both NG V2 and NG V3 are competitive and overall are preferred to KOCHEN especially as the problem size increases. NR heuristic is very competitive with NG V3 and the results indicate local improvement helps improve the solution quality.

Table 48, based on the new test set, yields interesting insights. First, among legacy heuristics, KOCHEN is no longer clearly preferred as the L – M M1 competes quite well on the variety of problems. The performance of TYPE degrades, as expected given the broad range of problems generated and the specificity of the TYPE heuristic. Each of the new gradient methods, NG V1, NG V2, and NG V3, compete well against the best legacy heuristics with NG V3 being better than NG V2 overall. NR(P) yielded better solutions compared to NG V3, as seen in Table 47 and 48, since the local improvement phase helped improve overall solution quality.

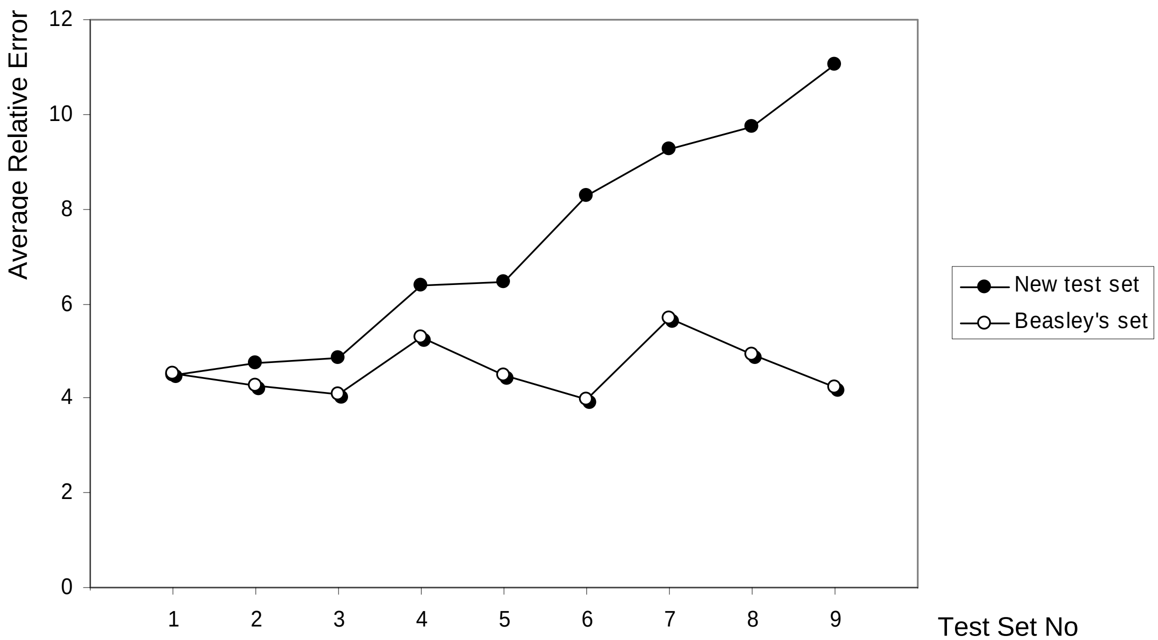


Figure 32. Average Relative Error of All Legacy Heuristics on Each Problem Test Set

The big difference, and one for which every researcher should take notice, arises in comparing the results in Tables 47 and 48. Notice in Table 48 that, for legacy heuristics, as the problem size increases, solution quality obtained degrades. This trend is not seen in Table 47 based on the current set of test problems. Figure 32 plots, by test problem grouping (there are 9 such groupings of 30 problems each), the average relative errors of all legacy heuristics; note the increasing trend associated with the new test set. This is not a favorable characteristic for a heuristic.

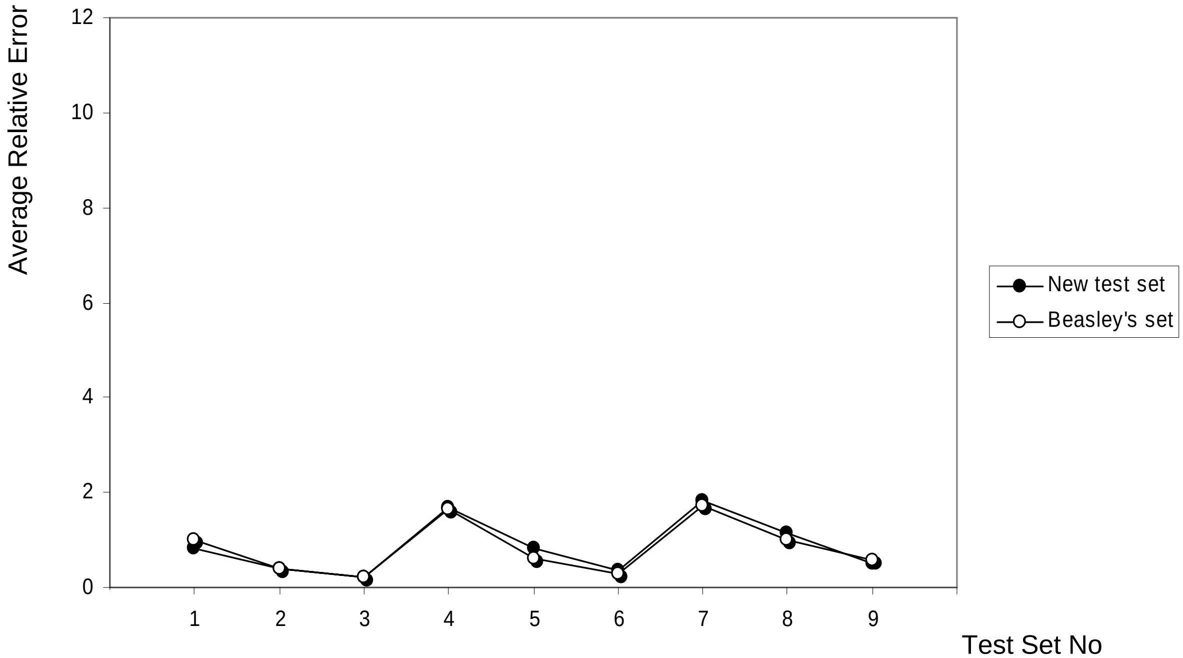


Figure 33. Average Relative Error of NG V3 on Each Problem Test Set

Figure 33 plots the average relative errors of NG V3 on each problem test set. In sharp contrast, the trends associated with the new heuristics developed in this research do not degrade in a similar fashion. NG V3 is especially robust which implies NG V3's greater potential utility for real-world applications.

VII. Conclusions

7.1 Contributions

This research employed an empirical science of heuristics applied to the multidimensional knapsack problem (MKP). Heuristics are often employed without deeply understanding solution performance. This research used the empirical analysis of the legacy greedy heuristics to gain a deeper knowledge of heuristic performance. Since existing standard problem sets do not include a sufficient range of problem characteristics, new test problem sets were developed for use in the computational experiments. The empirical testing uncovered the fact that different heuristics yield different solution qualities based on problem characteristics. The solution qualities are analyzed as a function of problem characteristics. Using this knowledge, several heuristics were defined, developed, and tested. These heuristics were found to yield robust and consistent solution quality over all problem characteristics.

7.1.1 Empirical Science Suggesting Theory

Hooker (1994) indicates that deductive mathematical methods are inadequate in studying the performance of algorithms based on computational experiment. Since understanding the influence of problem characteristics on heuristic solution performance requires computational testing, current deductive mathematical methods are inadequate. This research conducted computational testing, rather than using a mathematical deductive approach, to gain insights into how various heuristics function according to particular test problem characteristics. These insights yielded theories regarding heuristic procedure performance as a function of problem characteristics. These theories were exploited to yield new heuristics whose performance is more robust than that of the legacy heuristics.

7.1.2 Evidence of Lack of Diversity of the Existing Benchmark Test Set

Evidence was presented that the existing benchmark problem set and test sets randomly generated by researchers do not include a sufficient variety of problem characteristics. This research indicated that these narrow ranges of problem characteristics lead to incorrect conclusions regarding the solution performance of heuristics. Thus, benchmark problem sets do not provide the mechanism to gain the heuristic solution procedure performance insight with respect to problem characteristics attainable with the more diverse problem set eventually employed in this research.

7.1.3 Knowledge of Heuristic Performance Based on Problem Characteristics

Knowledge regarding heuristic solution performance was achieved using a structured empirical test of legacy heuristics using a more diverse test set of 2KP and 5KP test problems. The research analyzed why the heuristics yielded various solution qualities depending on problem characteristics, and uncovered new heuristic performance characteristics in terms of finding a best performer among the heuristics for particular problem types. Analysis of the rationale for these best performers yielded insights that led to new heuristic approaches.

7.1.4 New Robust Heuristics Development

Several new heuristics were developed based on the empirical analysis of legacy heuristics. The TYPE heuristic was based on pre-processing a problem to determine constraint slackness and correlation levels which facilitates the choice of heuristic likely to be the best performer. The TYPE heuristic, first suggested 30 years ago, is believed to be the first of its kind. Three new gradient heuristics, based on characteristics of legacy heuristics, were defined and developed. These new gradient heuristics performed extremely well in terms of returning best solutions, performing quite well over the range of constraint slackness and correlations. A new reduction heuristic showed the effectiveness of reducing the problem size using the core problem structure. The core problem reduction in this research was based on a gradient measure range versus total variable percentage and was shown adequate in terms of optimal value coverage. A local improvement method was devised and examined as well.

7.1.5 New Problem Generation Approach

Existing benchmark problems do not provide an adequate experimental basis upon which to make claims of general applicability of a heuristic solution procedure. This research proposed an alternative generation scheme to MKP generation. This problem generation approach allows constraint slackness and correlation levels to randomly vary within a constraint set, thereby yielding problems with diverse problem characteristics. A new test set was derived and includes a true range of problem characteristics. Performance results against this new test set demonstrated the deficiencies of legacy heuristics and the viability of the new heuristics developed.

7.2 Future Research

This research has focused on the solution procedure performance of greedy heuristics on the MKP. There are several areas that should be examined in future research:

First, existing modern heuristics may be affected by problem characteristics because the greedy heuristic produces various solution qualities depending on problem characteristics. Since new heuristics provide robust and consistent base solutions (initial solutions), modern heuristics, such as tabu search, genetic algorithms, simulated annealing, or an ant colony optimization algorithm, and their solution procedure performance could be examined using the new test set. This future research can identify which is the best modern heuristic approach for solving the MKP.

Second, future research could be applied to other problem types such as set covering problems, set partitioning problems, bin packing problems, and general assignment problems. Many combinatorial problems require a heuristic to solve them, and this research can improve the solution performance of heuristics and yield robust solution quality.

Third, future research could also examine existing benchmark problems of other problem types. If existing benchmark problem sets do not cover the full range of problem characteristics, this research could create sufficient problem generation methods for other problem types. These new problem sets could provide full information regarding heuristic solution performance.

Finally, this research applied just Glover's improvement phase (1977) and Pirkul's improvement phase (1987) to the MKP. These local improvements are not effective to improve the solution quality when they are combined with NG V3. Future research could consider improvement moves taken from meta-heuristics such as genetic algorithms and tabu search.

Appendix A. Statistical Test to Distinguish a Best Heuristic in 2KP

- Chi-square test

$\mathbf{H_0^C}$: Heuristics do not differ.

$\mathbf{H_1^C}$: At least one heuristic differs.

Table A.1 Chi - Square Test for 2KP Constraint Slackness

Constraint Slackness	X^2	df	Probability	Reject Region? (α = 0.1)
1,1	671.709	7	0.000	**Yes**
1,2	152.291	7	0.000	**Yes**
2,1	144.873	7	0.000	**Yes**
2,2	345.091	7	0.000	**Yes**

- Sign test for two heuristics

$\mathbf{H_0^S}$**:** Two heuristics statistically have the same performance.

$\mathbf{H_1^S}$: One heuristic has statistically better performance compared to another heuristic.

Table A.2 Sign Test for 2KP Constraint Slackness

Constraint Slackness	Best Heuristic	# of Better Solutions	vs. Heuristic	# of Better Solutions	Pr(X ≥ U)	Reject Region (α = 0.1)
1,1	KOCHEN	216	TOYODA	15	0.000	**Yes**
		229	S – T	11	0.000	**Yes**
		249	L – M M1	20	0.000	**Yes**
		210	FOX	59	0.000	**Yes**
1,2	S – T	202	TOYODA	21	0.000	**Yes**
		246	L – M M1	17	0.000	**Yes**
		187	FOX	46	0.000	**Yes**
		73	KOCHEN	55	0.066	**Yes**
2,1	S – T	192	TOYODA	21	0.000	**Yes**
		244	L – M M1	14	0.000	**Yes**
		180	FOX	54	0.000	**Yes**
		70	KOCHEN	61	0.242	**No**
2,2	KOCHEN	195	TOYODA	15	0.000	**Yes**
		200	S – T	13	0.000	**Yes**
		227	L – M M1	27	0.000	**Yes**
		157	FOX	85	0.000	**Yes**

U ~ the number times best heuristic is better than compared heuristic

Table A.3 Sign Test for 2KP Correlation Structures

CorrCA1	CorrCA2	CorrA1A2	Best	Best	TOYODA	Pr(X≥U)	Reject Ho	Best	S -T	Pr(X≥U)	Reject Ho	Best	L - M M1	Pr(X≥U)	Reject Ho	Best	FOX	Pr(X≥U)	Reject Ho	Best	KOCHEN	Pr(X≥U)	Reject Ho
2	2	2	KOCHEN	8	0	0.007	**Y**	8	0	0.007	**Y**	10	0	0.002	**Y**	7	2	0.091	**Y**				
2	1	1	FOX	10	0	0.002	**Y**	10	0	0.002	**Y**	9	1	0.013	**Y**					9	1	0.013	**Y**
2	0	0	FOX	10	0	0.002	**Y**	10	0	0.002	**Y**	8	2	0.057	**Y**					7	3	0.171	**Y**
2	-1	-1	FOX	10	0	0.002	**Y**	10	0	0.002	**Y**	10	0	0.002	**Y**					7	3	0.171	**Y**
2	-2	-2	FOX	8	1	0.023	**Y**	10	0	0.002	**Y**	10	0	0.002	**Y**					7	3	0.171	**Y**
1	2	1	FOX	10	0	0.002	**Y**	10	0	0.002	**Y**	9	1	0.013	**Y**					10	0	0.002	**Y**
1	1	2	FOX	7	3	0.171	**Y**	7	3	0.171	**Y**	10	0	0.002	**Y**					6	4	0.376	**N**
1	1	1	KOCHEN	7	3	0.171	**Y**	7	3	0.171	**Y**	20	0	0.000	**Y**	20	0	0.000	**Y**				
1	1	0	KOCHEN	6	1	0.065	**Y**	6	1	0.065	**Y**	10	0	0.002	**Y**	10	0	0.002	**Y**				
1	0	1	FOX	8	2	0.057	**Y**	9	1	0.013	**Y**	9	1	0.013	**Y**					6	4	0.376	**N**
1	0	0	KOCHEN	20	0	0.000	**Y**	20	0	0.000	**Y**	20	0	0.000	**Y**	19	1	0.000	**Y**				
1	0	-1	KOCHEN	10	0	0.002	**Y**	10	0	0.002	**Y**	10	0	0.002	**Y**	10	0	0.002	**Y**				
1	-1	0	FOX	10	0	0.002	**Y**	10	0	0.002	**Y**	10	0	0.002	**Y**	5	5	0.624	**N**				
1	-1	-1	KOCHEN	20	0	0.000	**Y**	20	0	0.000	**Y**	17	0	0.000	**Y**	16	3	0.003	**Y**				
1	-1	-2	KOCHEN	10	0	0.002	**Y**	10	0	0.002	**Y**	10	0	0.002	**Y**	10	0	0.002	**Y**				
1	-2	-1	KOCHEN	9	0	0.004	**Y**	10	0	0.002	**Y**	10	0	0.002	**Y**	5	2	0.225	**N**				
0	2	0	FOX	10	0	0.002	**Y**	10	0	0.002	**Y**	10	0	0.002	**Y**					9	1	0.013	**Y**
0	1	1	FOX	10	0	0.002	**Y**	10	0	0.002	**Y**	10	0	0.002	**Y**					6	2	0.144	**Y**
0	1	0	KOCHEN	20	0	0.000	**Y**	20	0	0.000	**Y**	19	1	0.000	**Y**	19	1	0.000	**Y**				
0	1	-1	KOCHEN	10	0	0.002	**Y**	10	0	0.002	**Y**	10	0	0.002	**Y**	10	0	0.002	**Y**				
0	0	2	TIE																				
0	0	1	KOCHEN	8	1	0.074	**Y**	8	1	0.031	**Y**	16	2	0.000	**Y**	20	0	0.000	**Y**				
0	0	0	KOCHEN	9	3	0.074	**Y**	11	3	0.031	**Y**	19	1	0.000	**Y**	19	1	0.000	**Y**				
0	0	-1	KOCHEN	9	3	0.074	**Y**	11	4	0.061	**Y**	20	0	0.000	**Y**	20	0	0.000	**Y**				
0	0	-2	KOCHEN	2	3	0.814	**N**	2	3	0.814	**N**	8	2	0.057	**Y**	10	0	0.002	**Y**				
0	-1	1	KOCHEN	7	0	0.012	**Y**	7	0	0.012	**Y**	10	0	0.002	**Y**	4	3	0.500	**N**				
0	-1	0	KOCHEN	19	0	0.000	**Y**	20	0	0.000	**Y**	15	4	0.011	**Y**	16	1	0.000	**Y**				
0	-1	-1	KOCHEN	7	1	0.039	**Y**	8	0	0.007	**Y**	6	3	0.252	**N**	9	1	0.013	**Y**				
0	-2	0	FOX	10	0	0.002	**Y**	10	0	0.002	**Y**	9	0	0.004	**Y**					4	3	0.500	**N**
-1	2	-1	FOX	10	0	0.002	**Y**	10	0	0.002	**Y**	10	0	0.002	**Y**					9	1	0.013	**Y**
-1	1	0	KOCHEN	10	0	0.002	**Y**	10	0	0.002	**Y**	10	0	0.002	**Y**	7	3	0.171	**Y**				
-1	1	-1	KOCHEN	20	0	0.000	**Y**	20	0	0.000	**Y**	18	2	0.000	**Y**	17	3	0.002	**Y**				
-1	1	-2	KOCHEN	10	0	0.002	**Y**	10	0	0.002	**Y**	7	2	0.091	**Y**	10	0	0.002	**Y**				
-1	0	1	FOX	7	3	0.171	**Y**	9	1	0.013	**Y**	10	0	0.002	**Y**					5	4	0.500	**N**
-1	0	0	KOCHEN	18	1	0.000	**Y**	18	1	0.000	**Y**	14	4	0.017	**Y**	17	1	0.000	**Y**				
-1	0	-1	KOCHEN	8	1	0.023	**Y**	9	0	0.004	**Y**	8	2	0.057	**Y**	10	0	0.002	**Y**				
-1	-1	2	TIE																				
-1	-1	1	KOCHEN	5	3	0.362	**N**	6	2	0.144	**Y**	14	3	0.008	**Y**	13	6	0.084	**Y**				
-1	-1	0	KOCHEN	2	5	0.935	**N**	4	3	0.500	**N**	13	3	0.012	**Y**	13	5	0.049	**Y**				
-1	-2	1	KOCHEN	7	1	0.039	**Y**	7	1	0.039	**Y**	7	1	0.039	**Y**	5	4	0.500	**N**				
-2	2	-2	FOX	10	0	0.002	**Y**	10	0	0.002	**Y**	10	0	0.002	**Y**					8	1	0.023	**Y**
-2	1	-1	KOCHEN	10	0	0.002	**Y**	10	0	0.002	**Y**	10	0	0.002	**Y**	6	2	0.144	**Y**				
-2	0	0	KOCHEN	7	1	0.039	**Y**	8	0	0.007	**Y**	8	1	0.023	**Y**	4	5	0.748	**N**				
-2	-1	1	KOCHEN	6	2	0.144	**Y**	6	2	0.144	**N**	4	2	0.342	**N**	4	4	0.638	**N**				
-2	-2	2	TIE																				

(Reject Region: $\alpha = 0.1$)

Appendix B. Supplementary Data for Generating 5KP Problem Set

Table B.1 Possible Combinations for Correlation Structure for 5KP

Comb.	CA^1	CA^2	CA^3	CA^4	CA^5	Comb.	CA^1	CA^2	CA^3	CA^4	CA^5	Comb.	CA^1	CA^2	CA^3	CA^4	CA^5
1	-2	-2	-2	-2	-2	**43**	-2	-1	-1	1	1	**85**	-1	-1	0	1	2
2	-2	-2	-2	-2	-1	**44**	-2	-1	-1	1	2	**86**	-1	-1	0	2	2
3	-2	-2	-2	-2	0	**45**	-2	-1	-1	2	2	**87**	-1	-1	1	1	1
4	-2	-2	-2	-2	1	**46**	-2	-1	0	0	0	**88**	-1	-1	1	1	2
5	-2	-2	-2	-2	2	**47**	-2	-1	0	0	1	**89**	-1	-1	1	2	2
6	-2	-2	-2	-1	-1	**48**	-2	-1	0	0	2	**90**	-1	-1	2	2	2
7	-2	-2	-2	-1	0	**49**	-2	-1	0	1	1	**91**	-1	0	0	0	0
8	-2	-2	-2	-1	1	**50**	-2	-1	0	1	2	**92**	-1	0	0	0	1
9	-2	-2	-2	-1	2	**51**	-2	-1	0	2	2	**93**	-1	0	0	0	2
10	-2	-2	-2	0	0	**52**	-2	-1	1	1	1	**94**	-1	0	0	1	1
11	-2	-2	-2	0	1	**53**	-2	-1	1	1	2	**95**	-1	0	0	1	2
12	-2	-2	-2	0	2	**54**	-2	-1	1	2	2	**96**	-1	0	0	2	2
13	-2	-2	-2	1	1	**55**	-2	-1	2	2	2	**97**	-1	0	1	1	1
14	-2	-2	-2	1	2	**56**	-2	0	0	0	0	**98**	-1	0	1	1	2
15	-2	-2	-2	2	2	**57**	-2	0	0	0	1	**99**	-1	0	1	2	2
16	-2	-2	-1	-1	-1	**58**	-2	0	0	0	2	**100**	-1	0	2	2	2
17	-2	-2	-1	-1	0	**59**	-2	0	0	1	1	**101**	-1	1	1	1	1
18	-2	-2	-1	-1	1	**60**	-2	0	0	1	2	**102**	-1	1	1	1	2
19	-2	-2	-1	-1	2	**61**	-2	0	0	2	2	**103**	-1	1	1	2	2
20	-2	-2	-1	0	0	**62**	-2	0	1	1	1	**104**	-1	1	2	2	2
21	-2	-2	-1	0	1	**63**	-2	0	1	1	2	**105**	-1	2	2	2	2
22	-2	-2	-1	0	2	**64**	-2	0	1	2	2	**106**	0	0	0	0	0
23	-2	-2	-1	1	1	**65**	-2	0	2	2	2	**107**	0	0	0	0	1
24	-2	-2	-1	1	2	**66**	-2	1	1	1	1	**108**	0	0	0	0	2
25	-2	-2	-1	2	2	**67**	-2	1	1	1	2	**109**	0	0	0	1	1
26	-2	-2	0	0	0	**68**	-2	1	1	2	2	**110**	0	0	0	1	2
27	-2	-2	0	0	1	**69**	-2	1	2	2	2	**111**	0	0	0	2	2
28	-2	-2	0	0	2	**70**	-2	2	2	2	2	**112**	0	0	1	1	1
29	-2	-2	0	1	1	**71**	-1	-1	-1	-1	-1	**113**	0	0	1	1	2
30	-2	-2	0	1	2	**72**	-1	-1	-1	-1	0	**114**	0	0	1	2	2
31	-2	-2	0	2	2	**73**	-1	-1	-1	-1	1	**115**	0	0	2	2	2
32	-2	-2	1	1	1	**74**	-1	-1	-1	-1	2	**116**	0	1	1	1	1
33	-2	-2	1	1	2	**75**	-1	-1	-1	0	0	**117**	0	1	1	1	2
34	-2	-2	1	2	2	**76**	-1	-1	-1	0	1	**118**	0	1	1	2	2
35	-2	-2	2	2	2	**77**	-1	-1	-1	0	2	**119**	0	1	2	2	2
36	-2	-1	-1	-1	-1	**78**	-1	-1	-1	1	1	**120**	0	2	2	2	2
37	-2	-1	-1	-1	0	**79**	-1	-1	-1	1	2	**121**	1	1	1	1	1
38	-2	-1	-1	-1	1	**80**	-1	-1	-1	2	2	**122**	1	1	1	1	2
39	-2	-1	-1	-1	2	**81**	-1	-1	0	0	0	**123**	1	1	1	2	2
40	-2	-1	-1	0	0	**82**	-1	-1	0	0	1	**124**	1	1	2	2	2
41	-2	-1	-1	0	1	**83**	-1	-1	0	0	2	**125**	1	2	2	2	2
42	-2	-1	-1	0	2	**84**	-1	-1	0	1	1	**126**	2	2	2	2	2

where CAi = ρ_{CA^i}, i = 1, …, 5, and coded correlation {-2, -1, 0, 1, 2} represents correlation { -0.9, -0.5, 0, 0.5, 0.9}.

Appendix C. Statistical Test to Distinguish a Best Heuristic in 5KP

- Chi-square test

$\mathbf{H_0^C}$: Heuristics do not differ.

$\mathbf{H_1^C}$: At least one heuristic differs.

Table C.1 Chi - Square Test for 5KP Constraint Slackness

Constraint Slackness	X^2	df	Probability	Reject Region? ($\alpha = 0.1$)
1,1,1,1,1	1069.210	4	0.000	Yes
1,1,1,1,2	1207.090	4	0.000	Yes
1,1,1,2,2	977.301	4	0.000	Yes
1,1,2,2,2	642.791	4	0.000	Yes
1,2,2,2,2	315.162	4	0.000	Yes
2,2,2,2,2	982.037	4	0.000	Yes

- Sign test for two heuristics

$\mathbf{H_0^S}$**:** Two heuristics statistically have the same performance.

$\mathbf{H_1^S}$: One heuristic has statistically better performance compared to another heuristic.

Table C.2 Sign Test for 5KP Constraint Slackness

Constraint Slackness	Best Heuristic	# of Better Solutions	vs. Heuristic	# of Better Solutions	$Pr(X \geq U)$	Reject Region ($\alpha = 0.1$)
1,1,1,1,1	KOCHEN	604	TOYODA	13	0.000	**Yes**
		618	S – T	7	0.000	**Yes**
		499	L – M M1	126	0.000	**Yes**
		554	FOX	75	0.000	**Yes**
1,1,1,1,2	KOCHEN	577	TOYODA	26	0.000	**Yes**
		599	S – T	14	0.000	**Yes**
		516	L – M M1	105	0.000	**Yes**
		583	FOX	45	0.000	**Yes**
1,1,1,2,2	KOCHEN	534	TOYODA	38	0.000	**Yes**
		553	S – T	37	0.000	**Yes**
		512	L – M M1	108	0.000	**Yes**
		603	FOX	27	0.000	**Yes**
1,1,2,2,2	KOCHEN	535	TOYODA	42	0.000	**Yes**
		456	S – T	80	0.000	**Yes**
		476	L – M M1	122	0.000	**Yes**
		606	FOX	22	0.000	**Yes**
1,2,2,2,2	S - T	451	TOYODA	155	0.000	**Yes**
		362	L – M M1	218	0.000	**Yes**
		591	FOX	19	0.000	**Yes**
		241	KOCHEN	237	0.445	**No**
2,2,2,2,2	KOCHEN	584	TOYODA	10	0.000	**Yes**
		594	S – T	6	0.000	**Yes**
		537	L – M M1	76	0.000	**Yes**
		499	FOX	123	0.000	**Yes**

U ~ the number times best heuristic is better than compared heuristic

Table C.3 Resource Usage in All Slackness Setting under Zero Correlation by Each Heuristic and Optimal Solution for 5KP

Slackness	Classification	TOYODA	S - T	L – M M1	FOX	KOCHEN	Optimal
(1,1,1,1,1)	# of Vars	34.0	33.8	32.4	34.0	34.0	34.0
	1st Const	0.963	0.968	0.984	0.984	0.962	0.988
	2nd Const	0.968	0.967	0.969	0.986	0.967	0.995
	3rd Const	0.965	0.947	0.974	0.970	0.956	0.995
	4th Const	0.970	0.964	0.977	0.972	0.982	0.991
	5th Const	0.982	0.974	0.986	0.982	0.974	0.993
(1,1,1,1,2)	# of Vars	34.4	34.2	32.8	35.8	35.0	35.0
	1st Const	0.973	0.973	0.987	0.972	0.983	0.990
	2nd Const	0.937	0.945	0.979	0.983	0.986	0.997
	3rd Const	0.983	0.978	0.987	0.980	0.975	0.995
	4th Const	0.960	0.943	0.975	0.973	0.976	0.994
	5th Const	0.478	0.478	0.468	0.512	0.500	0.506
(1,1,1,2,2)	# of Vars	35.2	35.2	34.6	37.0	36.6	36.6
	1st Const	0.946	0.940	0.988	0.993	0.993	0.994
	2nd Const	0.970	0.978	0.987	0.981	0.984	0.998
	3rd Const	0.941	0.923	0.993	0.979	0.978	0.989
	4th Const	0.466	0.474	0.489	0.507	0.503	0.513
	5th Const	0.492	0.505	0.495	0.525	0.525	0.527
(1,1,2,2,2)	# of Vars	38.4	39.4	36.4	40.200	40.2	39.4
	1st Const	0.976	0.977	0.988	0.994	0.987	0.998
	2nd Const	0.993	0.995	0.993	0.991	0.991	0.997
	3rd Const	0.516	0.548	0.510	0.566	0.556	0.564
	4th Const	0.530	0.555	0.543	0.587	0.574	0.565
	5th Const	0.522	0.556	0.521	0.551	0.557	0.556
(1,2,2,2,2)	# of Vars	40.8	44.2	42.2	45.6	44.6	45.6
	1st Const	0.998	0.998	0.996	0.992	0.997	1.000
	2nd Const	0.564	0.620	0.599	0.643	0.628	0.655
	3rd Const	0.564	0.625	0.613	0.654	0.629	0.659
	4th Const	0.542	0.603	0.593	0.638	0.612	0.624
	5th Const	0.550	0.613	0.615	0.647	0.629	0.658
(2,2,2,2,2)	# of Vars	70.4	70.4	69.4	72.4	70.8	71.2
	1st Const	0.968	0.968	0.982	0.986	0.976	0.985
	2nd Const	0.986	0.986	0.988	0.988	0.994	0.998
	3rd Const	0.984	0.984	0.986	0.985	0.985	0.996
	4th Const	0.974	0.974	0.985	0.993	0.987	0.995
	5th Const	0.985	0.985	0.988	0.983	0.988	0.997

Table C.4 Sign Tests for 5KP Correlation Structures

Correlation	Best	Best	TOYODA	Pr(X≥U)	Reject Ho	Best	S -T	Pr(X≥U)	Reject Ho	Best	L - M M1	Pr(X≥U)	Reject Ho	Best	FOX	Pr(X≥U)	Reject Ho	Best	KOCHEN	Pr(X≥U)	Reject Ho
-2,-2,-2,-2,-2	TIE																				
-2,-2,-2,-2,-1	KOCHEN	8	0	0.007	Y	8	0	0.007	Y	5	4	0.500	N	8	2	0.057	Y				
-2,-2,-2,-2,0	FOX	10	0	0.002	Y	10	0	0.002	Y	7	3	0.171	Y					8	1	0.023	Y
-2,-2,-2,-2,1	FOX	10	0	0.002	Y	10	0	0.002	Y	6	3	0.252	N					7	3	0.171	Y
-2,-2,-2,-2,2	FOX	10	0	0.002	Y	10	0	0.002	Y	9	1	0.013	Y					9	1	0.013	Y
-2,-2,-2,-1,-1	L-M M1	7	2	0.091	Y	8	1	0.023	Y					9	1	0.013	Y	4	4	0.638	N
-2,-2,-2,-1,0	L-M M1	10	0	0.002	Y	10	0	0.002	Y					8	2	0.057	Y	7	3	0.171	Y
-2,-2,-2,-1,1	L-M M1	10	0	0.002	Y	10	0	0.002	Y					6	4	0.376	N	6	4	0.376	N
-2,-2,-2,-1,2	FOX	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y					10	0	0.002	Y
-2,-2,-2,0,0	KOCHEN	9	1	0.013	Y	9	1	0.013	Y	10	0	0.002	Y	9	0	0.004	Y				
-2,-2,-2,0,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,-2,-2,0,2	FOX	10	0	0.002	Y	10	0	0.002	Y	9	1	0.013	Y					8	2	0.057	Y
-2,-2,-2,1,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,-2,-2,1,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	7	3	0.171	Y	6	4	0.376	N				
-2,-2,-2,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,-2,-1,-1,-1	KOCHEN	6	0	0.021	Y	7	0	0.012	Y	6	3	0.252	N	8	1	0.023	Y				
-2,-2,-1,-1,0	KOCHEN	9	0	0.004	Y	9	0	0.004	Y	4	5	0.748	N	9	1	0.013	Y				
-2,-2,-1,-1,1	L-M M1	10	0	0.002	Y	10	0	0.002	Y					6	4	0.376	N	7	3	0.171	Y
-2,-2,-1,-1,2	FOX	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y					8	0	0.007	Y
-2,-2,-1,0,0	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	9	0	0.004	Y	10	0	0.002	Y				
-2,-2,-1,0,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	6	4	0.376	N	10	0	0.002	Y				
-2,-2,-1,0,2	FOX	9	1	0.013	Y	10	0	0.002	Y	9	1	0.013	Y					8	2	0.057	Y
-2,-2,-1,1,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	9	0	0.004	Y	10	0	0.002	Y				
-2,-2,-1,1,2	TIE																				
-2,-2,-1,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,-2,0,0,0	KOCHEN	8	1	0.023	Y	9	0	0.004	Y	9	1	0.013	Y	10	0	0.002	Y				
-2,-2,0,0,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	8	2	0.057	Y	10	0	0.002	Y				
-2,-2,0,0,2	FOX	10	0	0.002	Y	10	0	0.002	Y	8	2	0.057	Y					5	5	0.624	N
-2,-2,0,1,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,-2,0,1,2	TIE																				
-2,-2,0,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,-2,1,1,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,-2,1,1,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	7	3	0.171	Y	10	0	0.002	Y				
-2,-2,1,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,-2,2,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,-1,-1,-1,-1	KOCHEN	6	1	0.065	Y	8	0	0.007	Y	5	5	0.624	N	10	0	0.002	Y				
-2,-1,-1,-1,0	L-M M1	10	0	0.002	Y	10	0	0.002	Y					10	0	0.002	Y	9	1	0.013	Y
-2,-1,-1,-1,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	7	3	0.171	Y	7	3	0.171	Y				
-2,-1,-1,-1,2	FOX	9	1	0.013	Y	10	0	0.002	Y	7	3	0.171	Y					8	2	0.057	Y
-2,-1,-1,0,0	KOCHEN	8	0	0.007	Y	10	0	0.002	Y	9	1	0.013	Y	10	0	0.002	Y				
-2,-1,-1,0,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	7	3	0.171	Y	10	0	0.002	Y				
-2,-1,-1,0,2	FOX	10	0	0.002	Y	10	0	0.002	Y	9	1	0.013	Y					9	1	0.013	Y
-2,-1,-1,1,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,-1,-1,1,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	8	2	0.057	Y	6	4	0.376	N				
-2,-1,-1,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	9	1	0.013	Y				
-2,-1,0,0,0	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	9	1	0.013	Y	10	0	0.002	Y				
-2,-1,0,0,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	7	3	0.171	Y	9	1	0.013	Y				
-2,-1,0,0,2	FOX	10	0	0.002	Y	10	0	0.002	Y	7	3	0.171	Y					8	2	0.057	Y
-2,-1,0,1,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,-1,0,1,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	9	1	0.013	Y	5	5	0.624	N				
-2,-1,0,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,-1,1,1,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,-1,1,1,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	7	3	0.171	Y	8	2	0.057	Y				
-2,-1,1,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	9	1	0.013	Y	10	0	0.002	Y				
-2,-1,2,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	9	1	0.013	Y	10	0	0.002	Y				
-2,0,0,0,0	KOCHEN	7	0	0.012	Y	7	1	0.039	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,0,0,0,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	6	3	0.252	N	10	0	0.002	Y				
-2,0,0,0,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	7	3	0.171	Y	6	4	0.376	N				
-2,0,0,1,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,0,0,1,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	8	2	0.057	Y	8	2	0.057	Y				
-2,0,0,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,0,1,1,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,0,1,1,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	7	3	0.171	Y	8	2	0.057	Y				
-2,0,1,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	9	1	0.013	Y	10	0	0.002	Y				
-2,0,2,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,1,1,1,1	KOCHEN	6	3	0.252	N	8	2	0.057	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,1,1,1,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	9	1	0.013	Y	10	0	0.002	Y				
-2,1,1,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,1,2,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-2,2,2,2,2	KOCHEN	8	2	0.057	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-1,-1,-1,-1,-1	TIE																				
-1,-1,-1,-1,0	L-M M1	9	1	0.013	Y	9	1	0.013	Y					8	1	0.023	Y	8	2	0.057	Y
-1,-1,-1,-1,1	TIE																				
-1,-1,-1,-1,2	FOX	10	0	0.002	Y	10	0	0.002	Y	7	3	0.171	Y					8	2	0.057	Y
-1,-1,-1,0,0	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	9	1	0.013	Y	10	0	0.002	Y				
-1,-1,-1,0,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	9	1	0.013	Y	9	0	0.004	Y				
-1,-1,-1,0,2	FOX	9	1	0.013	Y	10	0	0.002	Y	8	2	0.057	Y					7	3	0.171	Y
-1,-1,-1,1,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-1,-1,-1,1,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	8	2	0.057	Y	8	2	0.057	Y				
-1,-1,-1,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-1,-1,0,0,0	KOCHEN	9	0	0.004	Y	10	0	0.002	Y	6	2	0.144	Y	10	0	0.002	Y				
-1,-1,0,0,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	8	2	0.057	Y	8	2	0.057	Y				
-1,-1,0,0,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	8	2	0.057	Y	7	3	0.171	Y				
-1,-1,0,1,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-1,-1,0,1,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	9	1	0.013	Y	5	5	0.624	N				
-1,-1,0,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	9	1	0.013	Y				
-1,-1,1,1,1	KOCHEN	9	1	0.013	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-1,-1,1,1,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	7	3	0.171	Y	9	1	0.013	Y				
-1,-1,1,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-1,-1,2,2,2	KOCHEN	9	0	0.004	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-1,0,0,0,0	KOCHEN	7	2	0.091	Y	8	1	0.023	Y	10	0	0.002	Y	10	0	0.002	Y				
-1,0,0,0,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	6	4	0.376	N	10	0	0.002	Y				
-1,0,0,0,2	TIE																				
-1,0,0,1,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-1,0,0,1,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	7	3	0.171	Y	9	1	0.013	Y				
-1,0,0,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-1,0,1,1,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-1,0,1,1,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	9	1	0.013	Y	10	0	0.002	Y				
-1,0,1,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	8	2	0.057	Y	10	0	0.002	Y				
-1,0,2,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-1,1,1,1,1	KOCHEN	8	2	0.057	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-1,1,1,1,2	L-M M1	9	1	0.013	Y	9	1	0.013	Y					9	1	0.013	Y	5	5	0.624	N
-1,1,1,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	9	1	0.013	Y	10	0	0.002	Y				
-1,1,2,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
-1,2,2,2,2	KOCHEN	9	1	0.013	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
0,0,0,0,0	TIE																				
0,0,0,0,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	7	3	0.171	Y	10	0	0.002	Y				
0,0,0,0,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	8	2	0.057	Y	7	3	0.171	Y				
0,0,0,1,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	9	1	0.013	Y	10	0	0.002	Y				
0,0,0,1,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	8	2	0.057	Y	8	2	0.057	Y				
0,0,0,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
0,0,1,1,1	KOCHEN	9	1	0.013	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
0,0,1,1,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	7	3	0.171	Y	8	2	0.057	Y				
0,0,1,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	7	2	0.091	Y	10	0	0.002	Y				
0,0,2,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
0,1,1,1,1	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
0,1,1,1,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	9	1	0.013	Y	10	0	0.002	Y				
0,1,1,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
0,1,2,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
0,2,2,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
1,1,1,1,1	KOCHEN	7	1	0.039	Y	8	1	0.023	Y	10	0	0.002	Y	10	0	0.002	Y				
1,1,1,1,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	6	4	0.376	N	10	0	0.002	Y				
1,1,1,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	9	1	0.013	Y	10	0	0.002	Y				
1,1,2,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
1,2,2,2,2	KOCHEN	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y	10	0	0.002	Y				
2,2,2,2,2	KOCHEN	9	1	0.013	Y	8	2	0.057	Y	10	0	0.002	Y	10	0	0.002	Y				

(Reject Region: $\alpha = 0.1$)

Table C. 5 Resource Usage by KOCHEN under (1, 1, 1, 2, 2)

Correlation	1st Const.	2nd Const.	3rd Const.	4th Const.	5th Const.
-2,-2,-2,-2,-2	0.948	0.959	0.957	0.421	0.401
-2,-2,-2,-2,-1	0.963	0.948	0.939	0.405	0.561
-2,-2,-2,-2,0	0.921	0.946	0.979	0.424	0.704
-2,-2,-2,-2,1	0.948	0.947	0.978	0.431	0.871
-2,-2,-2,-2,2	0.940	0.940	0.950	0.409	0.985
-2,-2,-2,-1,-1	0.971	0.964	0.921	0.562	0.579
-2,-2,-2,-1,0	0.978	0.986	0.943	0.556	0.731
-2,-2,-2,-1,1	0.964	0.941	0.953	0.565	0.873
-2,-2,-2,-1,2	0.945	0.960	0.938	0.544	0.985
-2,-2,-2,0,0	0.964	0.930	0.975	0.714	0.699
-2,-2,-2,0,1	0.938	0.955	0.972	0.720	0.882
-2,-2,-2,0,2	0.963	0.916	0.958	0.716	0.993
-2,-2,-2,1,1	0.948	0.950	0.968	0.878	0.879
-2,-2,-2,1,2	0.938	0.960	0.968	0.862	0.990
-2,-2,-2,2,2	0.918	0.962	0.958	0.975	0.981
-2,-2,-1,-1,-1	0.920	0.921	0.998	0.533	0.519
-2,-2,-1,-1,0	0.904	0.875	0.990	0.533	0.664
-2,-2,-1,-1,1	0.882	0.910	0.998	0.537	0.814
-2,-2,-1,-1,2	0.856	0.888	0.994	0.531	0.939
-2,-2,-1,0,0	0.908	0.898	0.995	0.683	0.668
-2,-2,-1,0,1	0.903	0.898	0.996	0.665	0.805
-2,-2,-1,0,2	0.916	0.894	0.995	0.673	0.923
-2,-2,-1,1,1	0.866	0.933	0.990	0.817	0.821
-2,-2,-1,1,2	0.882	0.914	0.995	0.827	0.948
-2,-2,-1,2,2	0.895	0.880	0.994	0.918	0.931
-2,-2,0,0,0	0.786	0.788	0.998	0.586	0.592
-2,-2,0,0,1	0.864	0.870	0.999	0.616	0.729
-2,-2,0,0,2	0.852	0.888	0.998	0.609	0.834
-2,-2,0,1,1	0.869	0.862	0.997	0.726	0.722
-2,-2,0,1,2	0.910	0.907	0.998	0.727	0.835
-2,-2,0,2,2	0.868	0.890	0.998	0.818	0.825
-2,-2,1,1,1	0.820	0.795	1.000	0.587	0.614
-2,-2,1,1,2	0.678	0.709	1.000	0.588	0.673
-2,-2,1,2,2	0.746	0.752	1.000	0.700	0.698
-2,-2,2,2,2	0.664	0.684	1.000	0.524	0.509
-2,-1,-1,-1,-1	0.856	0.989	0.987	0.501	0.503
-2,-1,-1,-1,0	0.838	0.979	0.996	0.458	0.630
-2,-1,-1,-1,1	0.784	0.988	0.976	0.484	0.771
-2,-1,-1,-1,2	0.815	0.993	0.990	0.502	0.912
-2,-1,-1,0,0	0.805	0.980	0.970	0.627	0.629
-2,-1,-1,0,1	0.810	0.987	0.993	0.609	0.766
-2,-1,-1,0,2	0.891	0.991	0.973	0.634	0.927
-2,-1,-1,1,1	0.818	0.988	0.980	0.789	0.769
-2,-1,-1,1,2	0.831	0.991	0.988	0.788	0.912
-2,-1,-1,2,2	0.746	0.980	0.969	0.892	0.890
-2,-1,0,0,0	0.764	0.953	0.993	0.570	0.554
-2,-1,0,0,1	0.780	0.978	0.987	0.604	0.717
-2,-1,0,0,2	0.813	0.983	0.996	0.583	0.821
-2,-1,0,1,1	0.753	0.968	0.994	0.701	0.704
-2,-1,0,1,2	0.772	0.956	0.997	0.682	0.803
-2,-1,0,2,2	0.707	0.967	0.997	0.814	0.812
-2,-1,1,1,1	0.741	0.820	1.000	0.600	0.585
-2,-1,1,1,2	0.781	0.915	0.996	0.605	0.689
-2,-1,1,2,2	0.748	0.872	1.000	0.686	0.683
-2,-1,2,2,2	0.616	0.655	1.000	0.501	0.511
-2,0,0,0,0	0.731	0.991	0.989	0.520	0.521
-2,0,0,0,1	0.759	0.980	0.994	0.552	0.665
-2,0,0,0,2	0.690	0.986	0.991	0.534	0.764
-2,0,0,1,1	0.688	0.996	0.986	0.633	0.660
-2,0,0,1,2	0.749	0.991	0.985	0.658	0.769
-2,0,0,2,2	0.705	0.994	0.994	0.772	0.775
-2,0,1,1,1	0.745	0.981	0.997	0.599	0.598
-2,0,1,1,2	0.787	0.973	0.998	0.595	0.672
-2,0,1,2,2	0.735	0.985	0.997	0.687	0.690

Correlation	1st Const.	2nd Const.	3rd Const.	4th Const.	5th Const.
-2,0,2,2,2	0.635	0.743	1.000	0.522	0.522
-2,1,1,1,1	0.746	0.997	0.998	0.554	0.555
-2,1,1,1,2	0.821	0.996	0.995	0.557	0.648
-2,1,1,2,2	0.776	0.983	0.998	0.635	0.633
-2,1,2,2,2	0.606	0.864	1.000	0.512	0.512
-2,2,2,2,2	0.760	0.997	1.000	0.499	0.508
-1,-1,-1,-1,-1	0.972	0.987	0.975	0.476	0.478
-1,-1,-1,-1,0	0.976	0.964	0.980	0.468	0.622
-1,-1,-1,-1,1	0.969	0.986	0.974	0.463	0.757
-1,-1,-1,-1,2	0.971	0.972	0.973	0.452	0.912
-1,-1,-1,0,0	0.975	0.970	0.985	0.595	0.620
-1,-1,-1,0,1	0.961	0.955	0.976	0.643	0.748
-1,-1,-1,0,2	0.958	0.986	0.978	0.614	0.900
-1,-1,-1,1,1	0.966	0.973	0.983	0.752	0.765
-1,-1,-1,1,2	0.967	0.985	0.991	0.755	0.871
-1,-1,-1,2,2	0.991	0.990	0.965	0.902	0.869
-1,-1,0,0,0	0.947	0.960	0.996	0.577	0.557
-1,-1,0,0,1	0.961	0.942	0.997	0.572	0.687
-1,-1,0,0,2	0.937	0.953	0.996	0.588	0.814
-1,-1,0,1,1	0.932	0.965	0.997	0.702	0.690
-1,-1,0,1,2	0.959	0.970	0.996	0.678	0.787
-1,-1,0,2,2	0.964	0.946	0.998	0.810	0.814
-1,-1,1,1,1	0.863	0.857	1.000	0.564	0.592
-1,-1,1,1,2	0.863	0.875	1.000	0.588	0.680
-1,-1,1,2,2	0.902	0.908	0.999	0.704	0.684
-1,-1,2,2,2	0.695	0.697	1.000	0.513	0.523
-1,0,0,0,0	0.886	0.994	0.993	0.529	0.529
-1,0,0,0,1	0.902	0.996	0.992	0.543	0.645
-1,0,0,0,2	0.891	0.992	0.994	0.518	0.766
-1,0,0,1,1	0.930	0.990	0.991	0.638	0.668
-1,0,0,1,2	0.891	0.992	0.988	0.644	0.755
-1,0,0,2,2	0.911	0.992	0.996	0.776	0.755
-1,0,1,1,1	0.837	0.952	1.000	0.582	0.586
-1,0,1,1,2	0.921	0.971	1.000	0.594	0.664
-1,0,1,2,2	0.895	0.955	1.000	0.673	0.675
-1,0,2,2,2	0.762	0.770	1.000	0.512	0.508
-1,1,1,1,1	0.835	0.996	0.987	0.529	0.546
-1,1,1,1,2	0.767	0.991	0.997	0.537	0.627
-1,1,1,2,2	0.890	0.992	0.995	0.627	0.618
-1,1,2,2,2	0.700	0.880	1.000	0.517	0.517
-1,2,2,2,2	0.856	0.993	0.999	0.482	0.489
0,0,0,0,0	0.993	0.984	0.978	0.503	0.525
0,0,0,0,1	0.978	0.989	0.975	0.485	0.627
0,0,0,0,2	0.983	0.984	0.985	0.493	0.722
0,0,0,1,1	0.991	0.983	0.981	0.637	0.636
0,0,0,1,2	0.981	0.994	0.994	0.608	0.722
0,0,0,2,2	0.971	0.988	0.991	0.743	0.726
0,0,1,1,1	0.961	0.947	0.998	0.554	0.581
0,0,1,1,2	0.951	0.975	0.996	0.579	0.659
0,0,1,2,2	0.973	0.969	0.993	0.651	0.663
0,0,2,2,2	0.753	0.784	1.000	0.505	0.511
0,1,1,1,1	0.908	0.998	0.991	0.537	0.527
0,1,1,1,2	0.910	0.995	0.987	0.534	0.605
0,1,1,2,2	0.957	0.997	0.996	0.631	0.626
0,1,2,2,2	0.846	0.947	1.000	0.519	0.517
0,2,2,2,2	0.838	0.998	0.998	0.502	0.494
1,1,1,1,1	0.988	0.981	0.990	0.544	0.535
1,1,1,1,2	0.993	0.969	0.996	0.540	0.612
1,1,1,2,2	0.994	0.996	0.993	0.605	0.600
1,1,2,2,2	0.842	0.870	0.999	0.493	0.482
1,2,2,2,2	0.875	0.993	0.998	0.470	0.485
2,2,2,2,2	0.993	0.997	0.994	0.481	0.492

Table C.6 Resource Usage by KOCHEN under (1, 1, 2, 2, 2)

Correlation	1st Const.	2nd Const.	3rd Const.	4th Const.	5th Const.
-2,-2,-2,-2,-2	0.945	0.951	0.441	0.442	0.438
-2,-2,-2,-2,-1	0.950	0.980	0.414	0.411	0.578
-2,-2,-2,-2,0	0.977	0.970	0.427	0.438	0.736
-2,-2,-2,-2,1	0.955	0.985	0.427	0.433	0.876
-2,-2,-2,-2,2	0.980	0.974	0.429	0.436	0.991
-2,-2,-2,-1,-1	0.974	0.960	0.446	0.577	0.562
-2,-2,-2,-1,0	0.948	0.963	0.456	0.589	0.740
-2,-2,-2,-1,1	0.959	0.984	0.449	0.588	0.887
-2,-2,-2,-1,2	0.956	0.946	0.457	0.571	0.989
-2,-2,-2,0,0	0.972	0.980	0.438	0.763	0.738
-2,-2,-2,0,1	0.960	0.982	0.441	0.716	0.889
-2,-2,-2,0,2	0.971	0.966	0.446	0.720	0.995
-2,-2,-2,1,1	0.991	0.951	0.435	0.889	0.876
-2,-2,-2,1,2	0.977	0.916	0.415	0.856	0.988
-2,-2,-2,2,2	0.952	0.939	0.433	0.980	0.986
-2,-2,-1,-1,-1	0.943	0.970	0.568	0.536	0.573
-2,-2,-1,-1,0	0.975	0.979	0.578	0.576	0.745
-2,-2,-1,-1,1	0.983	0.937	0.574	0.565	0.884
-2,-2,-1,-1,2	0.948	0.972	0.546	0.549	0.990
-2,-2,-1,0,0	0.936	0.979	0.590	0.731	0.725
-2,-2,-1,0,1	0.986	0.975	0.579	0.725	0.887
-2,-2,-1,0,2	0.985	0.941	0.562	0.727	0.983
-2,-2,-1,1,1	0.961	0.976	0.582	0.878	0.888
-2,-2,-1,1,2	0.980	0.975	0.569	0.876	0.992
-2,-2,-1,2,2	0.981	0.959	0.576	0.994	0.994
-2,-2,0,0,0	0.986	0.976	0.720	0.727	0.711
-2,-2,0,0,1	0.966	0.977	0.703	0.706	0.859
-2,-2,0,0,2	0.970	0.969	0.723	0.734	0.998
-2,-2,0,1,1	0.962	0.946	0.705	0.869	0.876
-2,-2,0,1,2	0.962	0.957	0.733	0.881	0.993
-2,-2,0,2,2	0.966	0.966	0.723	0.990	0.989
-2,-2,1,1,1	0.970	0.908	0.859	0.869	0.854
-2,-2,1,1,2	0.968	0.965	0.864	0.881	0.988
-2,-2,1,2,2	0.985	0.943	0.860	0.986	0.992
-2,-2,2,2,2	0.932	0.969	0.983	0.984	0.975
-2,-1,-1,-1,-1	0.949	0.998	0.547	0.544	0.530
-2,-1,-1,-1,0	0.893	0.994	0.532	0.522	0.681
-2,-1,-1,-1,1	0.862	0.996	0.507	0.489	0.793
-2,-1,-1,-1,2	0.937	0.995	0.520	0.543	0.919
-2,-1,-1,0,0	0.880	0.996	0.525	0.687	0.652
-2,-1,-1,0,1	0.918	0.996	0.532	0.665	0.819
-2,-1,-1,0,2	0.922	0.992	0.554	0.680	0.972
-2,-1,-1,1,1	0.933	0.996	0.540	0.801	0.803
-2,-1,-1,1,2	0.934	0.996	0.524	0.807	0.962
-2,-1,-1,2,2	0.888	0.989	0.532	0.939	0.954
-2,-1,0,0,0	0.931	0.999	0.652	0.662	0.677
-2,-1,0,0,1	0.911	0.993	0.643	0.663	0.813
-2,-1,0,0,2	0.896	0.988	0.656	0.651	0.942
-2,-1,0,1,1	0.898	0.992	0.665	0.803	0.807
-2,-1,0,1,2	0.897	0.996	0.682	0.833	0.950
-2,-1,0,2,2	0.907	0.995	0.650	0.912	0.924
-2,-1,1,1,1	0.895	0.995	0.844	0.823	0.830
-2,-1,1,1,2	0.903	0.995	0.805	0.801	0.952
-2,-1,1,2,2	0.895	0.996	0.818	0.949	0.955
-2,-1,2,2,2	0.936	0.994	0.956	0.958	0.962
-2,0,0,0,0	0.825	0.996	0.618	0.598	0.592
-2,0,0,0,1	0.895	0.998	0.605	0.614	0.746
-2,0,0,0,2	0.906	1.000	0.608	0.611	0.840
-2,0,0,1,1	0.927	0.997	0.609	0.706	0.713
-2,0,0,1,2	0.907	0.998	0.593	0.741	0.844
-2,0,0,2,2	0.879	0.998	0.595	0.832	0.840
-2,0,1,1,1	0.876	0.998	0.706	0.712	0.714
-2,0,1,1,2	0.894	0.999	0.731	0.740	0.837
-2,0,1,2,2	0.904	0.995	0.730	0.845	0.849

Correlation	1st Const.	2nd Const.	3rd Const.	4th Const.	5th Const.
-2,0,2,2,2	0.859	0.999	0.832	0.821	0.822
-2,1,1,1,1	0.846	0.999	0.624	0.611	0.627
-2,1,1,1,2	0.869	0.998	0.599	0.615	0.697
-2,1,1,2,2	0.914	0.997	0.622	0.712	0.711
-2,1,2,2,2	0.927	0.998	0.710	0.695	0.702
-2,2,2,2,2	0.804	1.000	0.538	0.534	0.530
-1,-1,-1,-1,-1	0.993	0.987	0.472	0.485	0.489
-1,-1,-1,-1,0	0.992	0.987	0.495	0.497	0.638
-1,-1,-1,-1,1	0.975	0.992	0.476	0.491	0.766
-1,-1,-1,-1,2	0.983	0.992	0.504	0.524	0.917
-1,-1,-1,0,0	0.988	0.988	0.513	0.655	0.626
-1,-1,-1,0,1	0.983	0.988	0.503	0.635	0.762
-1,-1,-1,0,2	0.989	0.983	0.514	0.672	0.915
-1,-1,-1,1,1	0.985	0.987	0.498	0.783	0.785
-1,-1,-1,1,2	0.980	0.985	0.525	0.787	0.906
-1,-1,-1,2,2	0.985	0.991	0.484	0.892	0.903
-1,-1,0,0,0	0.979	0.989	0.654	0.643	0.652
-1,-1,0,0,1	0.992	0.982	0.643	0.626	0.771
-1,-1,0,0,2	0.988	0.980	0.656	0.658	0.923
-1,-1,0,1,1	0.987	0.984	0.642	0.758	0.785
-1,-1,0,1,2	0.991	0.991	0.651	0.791	0.913
-1,-1,0,2,2	0.982	0.990	0.644	0.913	0.930
-1,-1,1,1,1	0.979	0.977	0.777	0.792	0.794
-1,-1,1,1,2	0.990	0.983	0.770	0.776	0.902
-1,-1,1,2,2	0.991	0.977	0.794	0.907	0.923
-1,-1,2,2,2	0.984	0.988	0.914	0.910	0.908
-1,0,0,0,0	0.969	0.993	0.574	0.560	0.574
-1,0,0,0,1	0.976	0.994	0.608	0.594	0.733
-1,0,0,0,2	0.980	0.995	0.585	0.580	0.817
-1,0,0,1,1	0.978	0.996	0.596	0.725	0.729
-1,0,0,1,2	0.955	0.998	0.570	0.687	0.813
-1,0,0,2,2	0.985	0.987	0.598	0.850	0.830
-1,0,1,1,1	0.964	0.994	0.701	0.714	0.696
-1,0,1,1,2	0.973	0.995	0.698	0.700	0.829
-1,0,1,2,2	0.975	0.997	0.704	0.816	0.800
-1,0,2,2,2	0.969	0.995	0.814	0.804	0.813
-1,1,1,1,1	0.896	0.999	0.582	0.595	0.592
-1,1,1,1,2	0.924	0.998	0.604	0.621	0.696
-1,1,1,2,2	0.902	0.999	0.608	0.688	0.692
-1,1,2,2,2	0.958	0.999	0.707	0.701	0.695
-1,2,2,2,2	0.807	1.000	0.551	0.544	0.540
0,0,0,0,0	0.987	0.991	0.556	0.574	0.557
0,0,0,0,1	0.986	0.992	0.557	0.544	0.666
0,0,0,0,2	0.984	0.994	0.571	0.563	0.781
0,0,0,1,1	0.990	0.985	0.528	0.634	0.644
0,0,0,1,2	0.991	0.993	0.565	0.676	0.752
0,0,0,2,2	0.987	0.992	0.557	0.763	0.786
0,0,1,1,1	0.998	0.988	0.643	0.652	0.645
0,0,1,1,2	0.993	0.987	0.648	0.655	0.777
0,0,1,2,2	0.989	0.991	0.646	0.767	0.774
0,0,2,2,2	0.994	0.996	0.776	0.776	0.778
0,1,1,1,1	0.978	0.997	0.592	0.599	0.594
0,1,1,1,2	0.982	0.997	0.605	0.612	0.692
0,1,1,2,2	0.960	0.996	0.591	0.675	0.658
0,1,2,2,2	0.979	0.995	0.690	0.683	0.671
0,2,2,2,2	0.801	1.000	0.517	0.525	0.512
1,1,1,1,1	0.995	0.992	0.564	0.563	0.552
1,1,1,1,2	0.996	0.992	0.576	0.578	0.642
1,1,1,2,2	0.997	0.995	0.577	0.629	0.638
1,1,2,2,2	0.996	0.993	0.625	0.619	0.649
1,2,2,2,2	0.880	1.000	0.509	0.514	0.515
2,2,2,2,2	0.994	0.998	0.506	0.515	0.517

Table C.7 Resource Usage by KOCHEN under (1, 2, 2, 2, 2)

Correlation	1st Const.	2nd Const.	3rd Const.	4th Const.	5th Const.
-2,-2,-2,-2,-2	0.992	0.474	0.467	0.478	0.492
-2,-2,-2,-2,-1	0.988	0.461	0.490	0.472	0.613
-2,-2,-2,-2,0	0.988	0.433	0.438	0.431	0.732
-2,-2,-2,-2,1	0.993	0.502	0.476	0.467	0.912
-2,-2,-2,-2,2	0.979	0.474	0.434	0.466	0.990
-2,-2,-2,-1,-1	0.992	0.452	0.467	0.609	0.608
-2,-2,-2,-1,0	0.993	0.476	0.474	0.609	0.766
-2,-2,-2,-1,1	0.991	0.475	0.483	0.603	0.903
-2,-2,-2,-1,2	0.984	0.423	0.437	0.588	0.995
-2,-2,-2,0,0	0.980	0.474	0.492	0.749	0.763
-2,-2,-2,0,1	0.996	0.433	0.469	0.740	0.880
-2,-2,-2,0,2	0.990	0.470	0.471	0.722	0.991
-2,-2,-2,1,1	0.982	0.426	0.440	0.880	0.885
-2,-2,-2,1,2	0.981	0.445	0.451	0.874	0.996
-2,-2,-2,2,2	0.989	0.440	0.428	0.985	0.986
-2,-2,-1,-1,-1	0.989	0.465	0.604	0.609	0.609
-2,-2,-1,-1,0	0.993	0.450	0.595	0.586	0.767
-2,-2,-1,-1,1	0.991	0.466	0.601	0.601	0.908
-2,-2,-1,-1,2	0.978	0.427	0.578	0.565	0.996
-2,-2,-1,0,0	0.992	0.462	0.601	0.757	0.777
-2,-2,-1,0,1	0.991	0.457	0.575	0.722	0.900
-2,-2,-1,0,2	0.973	0.482	0.598	0.732	0.992
-2,-2,-1,1,1	0.996	0.440	0.588	0.887	0.893
-2,-2,-1,1,2	0.981	0.438	0.561	0.868	0.994
-2,-2,-1,2,2	0.988	0.479	0.562	0.992	0.995
-2,-2,0,0,0	0.988	0.449	0.741	0.730	0.745
-2,-2,0,0,1	0.989	0.463	0.764	0.760	0.919
-2,-2,0,0,2	0.979	0.431	0.718	0.703	0.992
-2,-2,0,1,1	0.989	0.456	0.740	0.884	0.877
-2,-2,0,1,2	0.981	0.435	0.728	0.876	0.990
-2,-2,0,2,2	0.996	0.451	0.724	0.987	0.996
-2,-2,1,1,1	0.995	0.455	0.913	0.894	0.899
-2,-2,1,1,2	0.985	0.464	0.895	0.900	0.992
-2,-2,1,2,2	0.986	0.444	0.860	0.987	0.988
-2,-2,2,2,2	0.982	0.454	0.989	0.991	0.986
-2,-1,-1,-1,-1	0.997	0.613	0.610	0.612	0.602
-2,-1,-1,-1,0	0.990	0.638	0.619	0.602	0.747
-2,-1,-1,-1,1	0.989	0.565	0.561	0.567	0.867
-2,-1,-1,-1,2	0.975	0.578	0.578	0.590	0.997
-2,-1,-1,0,0	0.991	0.623	0.614	0.765	0.760
-2,-1,-1,0,1	0.992	0.597	0.581	0.738	0.872
-2,-1,-1,0,2	0.990	0.565	0.558	0.710	0.981
-2,-1,-1,1,1	0.990	0.590	0.616	0.897	0.894
-2,-1,-1,1,2	0.987	0.580	0.586	0.872	0.996
-2,-1,-1,2,2	0.978	0.587	0.563	0.992	0.984
-2,-1,0,0,0	0.992	0.612	0.751	0.743	0.777
-2,-1,0,0,1	0.987	0.601	0.740	0.759	0.908
-2,-1,0,0,2	0.984	0.584	0.721	0.731	0.993
-2,-1,0,1,1	0.992	0.590	0.746	0.896	0.902
-2,-1,0,1,2	0.982	0.577	0.736	0.875	0.996
-2,-1,0,2,2	0.986	0.562	0.709	0.985	0.995
-2,-1,1,1,1	0.985	0.608	0.894	0.871	0.896
-2,-1,1,1,2	0.975	0.598	0.886	0.888	0.998
-2,-1,1,2,2	0.966	0.582	0.871	0.991	0.994
-2,-1,2,2,2	0.989	0.572	0.982	0.983	0.984
-2,0,0,0,0	0.992	0.762	0.743	0.774	0.760
-2,0,0,0,1	0.989	0.739	0.752	0.750	0.895
-2,0,0,0,2	0.986	0.729	0.726	0.713	0.986
-2,0,0,1,1	0.991	0.738	0.745	0.899	0.891
-2,0,0,1,2	0.978	0.721	0.728	0.872	0.995
-2,0,0,2,2	0.980	0.723	0.729	0.994	0.988
-2,0,1,1,1	0.993	0.716	0.881	0.881	0.885
-2,0,1,1,2	0.975	0.723	0.868	0.864	0.996
-2,0,1,2,2	0.982	0.728	0.883	0.989	0.987

Correlation	1st Const.	2nd Const.	3rd Const.	4th Const.	5th Const.
-2,0,2,2,2	0.979	0.734	0.989	0.994	0.989
-2,1,1,1,1	0.987	0.888	0.908	0.900	0.898
-2,1,1,1,2	0.985	0.866	0.879	0.890	0.992
-2,1,1,2,2	0.976	0.856	0.886	0.988	0.989
-2,1,2,2,2	0.987	0.837	0.980	0.974	0.967
-2,2,2,2,2	0.990	0.995	0.986	0.986	0.980
-1,-1,-1,-1,-1	0.996	0.586	0.565	0.569	0.579
-1,-1,-1,-1,0	0.994	0.567	0.562	0.546	0.688
-1,-1,-1,-1,1	0.996	0.560	0.566	0.536	0.836
-1,-1,-1,-1,2	0.995	0.573	0.560	0.569	0.949
-1,-1,-1,0,0	0.997	0.573	0.547	0.688	0.675
-1,-1,-1,0,1	0.994	0.565	0.582	0.695	0.842
-1,-1,-1,0,2	0.992	0.559	0.580	0.720	0.954
-1,-1,-1,1,1	0.992	0.550	0.572	0.830	0.835
-1,-1,-1,1,2	0.993	0.571	0.558	0.820	0.958
-1,-1,-1,2,2	0.997	0.561	0.555	0.962	0.944
-1,-1,0,0,0	0.992	0.549	0.703	0.688	0.679
-1,-1,0,0,1	0.996	0.560	0.696	0.716	0.827
-1,-1,0,0,2	0.993	0.608	0.710	0.716	0.969
-1,-1,0,1,1	0.995	0.567	0.684	0.827	0.826
-1,-1,0,1,2	0.993	0.571	0.682	0.813	0.949
-1,-1,0,2,2	0.993	0.590	0.714	0.968	0.963
-1,-1,1,1,1	0.997	0.543	0.830	0.826	0.838
-1,-1,1,1,2	0.996	0.589	0.835	0.841	0.971
-1,-1,1,2,2	0.993	0.543	0.807	0.938	0.929
-1,-1,2,2,2	0.999	0.593	0.952	0.944	0.946
-1,0,0,0,0	0.995	0.709	0.716	0.712	0.699
-1,0,0,0,1	0.996	0.687	0.705	0.696	0.823
-1,0,0,0,2	0.995	0.726	0.721	0.711	0.973
-1,0,0,1,1	0.995	0.688	0.696	0.812	0.817
-1,0,0,1,2	0.992	0.716	0.682	0.806	0.930
-1,0,0,2,2	0.992	0.704	0.700	0.951	0.954
-1,0,1,1,1	0.998	0.707	0.807	0.825	0.836
-1,0,1,1,2	0.995	0.685	0.829	0.790	0.937
-1,0,1,2,2	0.997	0.716	0.822	0.955	0.955
-1,0,2,2,2	0.993	0.671	0.944	0.945	0.924
-1,1,1,1,1	0.991	0.816	0.809	0.829	0.832
-1,1,1,1,2	0.994	0.834	0.832	0.836	0.952
-1,1,1,2,2	0.993	0.827	0.824	0.949	0.945
-1,1,2,2,2	0.992	0.834	0.947	0.966	0.938
-1,2,2,2,2	0.999	0.955	0.948	0.961	0.950
0,0,0,0,0	0.997	0.628	0.629	0.612	0.629
0,0,0,0,1	0.999	0.636	0.620	0.654	0.742
0,0,0,0,2	0.998	0.598	0.627	0.613	0.843
0,0,0,1,1	0.999	0.632	0.633	0.737	0.733
0,0,0,1,2	0.998	0.632	0.633	0.735	0.822
0,0,0,2,2	0.995	0.680	0.695	0.878	0.875
0,0,1,1,1	0.998	0.642	0.738	0.747	0.746
0,0,1,1,2	0.997	0.624	0.730	0.738	0.843
0,0,1,2,2	0.999	0.646	0.758	0.851	0.854
0,0,2,2,2	0.998	0.658	0.854	0.859	0.857
0,1,1,1,1	0.999	0.749	0.742	0.752	0.737
0,1,1,1,2	0.998	0.753	0.745	0.756	0.842
0,1,1,2,2	0.996	0.762	0.758	0.836	0.838
0,1,2,2,2	0.997	0.722	0.833	0.839	0.823
0,2,2,2,2	0.997	0.841	0.837	0.858	0.836
1,1,1,1,1	0.999	0.657	0.667	0.651	0.646
1,1,1,1,2	0.999	0.653	0.632	0.638	0.712
1,1,1,2,2	0.998	0.637	0.642	0.703	0.714
1,1,2,2,2	0.997	0.664	0.721	0.726	0.715
1,2,2,2,2	0.999	0.718	0.732	0.715	0.727
2,2,2,2,2	0.999	0.546	0.563	0.555	0.550

Table C. 8 Resource Usage by KOCHEN under (2, 2, 2, 2, 2)

Correlation	1st Const.	2nd Const.	3rd Const.	4th Const.	5th Const.
-2,-2,-2,-2,-2	0.961	0.979	0.971	0.978	0.973
-2,-2,-2,-2,-1	0.925	0.924	0.929	0.923	0.998
-2,-2,-2,-2,0	0.869	0.861	0.876	0.876	1.000
-2,-2,-2,-2,1	0.788	0.797	0.789	0.790	0.999
-2,-2,-2,-2,2	0.653	0.639	0.647	0.643	1.000
-2,-2,-2,-1,-1	0.888	0.871	0.908	0.980	0.993
-2,-2,-2,-1,0	0.834	0.851	0.845	0.941	0.998
-2,-2,-2,-1,1	0.775	0.779	0.766	0.873	1.000
-2,-2,-2,-1,2	0.592	0.602	0.586	0.672	1.000
-2,-2,-2,0,0	0.833	0.843	0.826	0.993	0.998
-2,-2,-2,0,1	0.720	0.726	0.735	0.952	1.000
-2,-2,-2,0,2	0.624	0.644	0.643	0.826	1.000
-2,-2,-2,1,1	0.701	0.717	0.710	0.995	0.999
-2,-2,-2,1,2	0.613	0.611	0.605	0.928	1.000
-2,-2,-2,2,2	0.630	0.645	0.625	0.998	0.999
-2,-2,-1,-1,-1	0.908	0.899	0.990	0.988	0.990
-2,-2,-1,-1,0	0.873	0.834	0.950	0.938	0.997
-2,-2,-1,-1,1	0.737	0.757	0.837	0.863	1.000
-2,-2,-1,-1,2	0.638	0.644	0.731	0.746	1.000
-2,-2,-1,0,0	0.798	0.778	0.889	0.991	0.994
-2,-2,-1,0,1	0.736	0.730	0.829	0.964	1.000
-2,-2,-1,0,2	0.573	0.590	0.701	0.802	1.000
-2,-2,-1,1,1	0.700	0.727	0.821	0.997	0.997
-2,-2,-1,1,2	0.642	0.634	0.743	0.914	1.000
-2,-2,-1,2,2	0.644	0.671	0.721	0.998	0.999
-2,-2,0,0,0	0.779	0.779	0.993	0.992	0.994
-2,-2,0,0,1	0.724	0.762	0.955	0.951	1.000
-2,-2,0,0,2	0.625	0.646	0.853	0.804	1.000
-2,-2,0,1,1	0.676	0.701	0.911	0.996	0.999
-2,-2,0,1,2	0.613	0.609	0.808	0.939	1.000
-2,-2,0,2,2	0.649	0.630	0.827	0.993	0.999
-2,-2,1,1,1	0.671	0.681	0.996	0.996	0.994
-2,-2,1,1,2	0.585	0.605	0.906	0.913	1.000
-2,-2,1,2,2	0.610	0.610	0.912	0.998	0.997
-2,-2,2,2,2	0.723	0.714	0.996	0.996	0.997
-2,-1,-1,-1,-1	0.886	0.987	0.973	0.976	0.983
-2,-1,-1,-1,0	0.825	0.907	0.920	0.905	0.999
-2,-1,-1,-1,1	0.762	0.868	0.881	0.858	1.000
-2,-1,-1,-1,2	0.620	0.709	0.709	0.727	1.000
-2,-1,-1,0,0	0.777	0.900	0.906	0.993	0.998
-2,-1,-1,0,1	0.746	0.835	0.848	0.959	1.000
-2,-1,-1,0,2	0.605	0.682	0.695	0.808	1.000
-2,-1,-1,1,1	0.732	0.813	0.830	0.998	0.998
-2,-1,-1,1,2	0.645	0.713	0.727	0.927	1.000
-2,-1,-1,2,2	0.613	0.711	0.698	0.993	0.999
-2,-1,0,0,0	0.822	0.916	0.995	0.987	0.995
-2,-1,0,0,1	0.722	0.858	0.953	0.942	1.000
-2,-1,0,0,2	0.623	0.711	0.805	0.821	1.000
-2,-1,0,1,1	0.696	0.823	0.934	0.995	0.999
-2,-1,0,1,2	0.641	0.736	0.831	0.938	1.000
-2,-1,0,2,2	0.608	0.695	0.796	0.999	1.000
-2,-1,1,1,1	0.704	0.800	0.997	0.993	0.997
-2,-1,1,1,2	0.614	0.706	0.914	0.925	1.000
-2,-1,1,2,2	0.609	0.703	0.919	0.998	0.999
-2,-1,2,2,2	0.623	0.722	0.991	0.999	0.996
-2,0,0,0,0	0.797	0.995	0.992	0.994	0.989
-2,0,0,0,1	0.719	0.928	0.946	0.948	1.000
-2,0,0,0,2	0.642	0.827	0.818	0.819	1.000
-2,0,0,1,1	0.729	0.937	0.939	0.999	0.998
-2,0,0,1,2	0.620	0.808	0.824	0.927	1.000
-2,0,0,2,2	0.649	0.824	0.820	0.999	0.998
-2,0,1,1,1	0.663	0.909	0.997	0.996	0.998
-2,0,1,1,2	0.644	0.835	0.919	0.923	1.000
-2,0,1,2,2	0.669	0.838	0.910	0.998	1.000

Correlation	1st Const.	2nd Const.	3rd Const.	4th Const.	5th Const.
-2,0,2,2,2	0.639	0.817	0.996	0.998	0.996
-2,1,1,1,1	0.720	0.997	0.993	0.996	0.991
-2,1,1,1,2	0.620	0.922	0.920	0.914	1.000
-2,1,1,2,2	0.693	0.936	0.918	1.000	0.999
-2,1,2,2,2	0.655	0.915	0.998	0.999	0.995
-2,2,2,2,2	0.760	0.996	0.996	0.993	0.995
-1,-1,-1,-1,-1	0.981	0.981	0.992	0.979	0.995
-1,-1,-1,-1,0	0.924	0.950	0.931	0.947	0.999
-1,-1,-1,-1,1	0.877	0.884	0.877	0.877	1.000
-1,-1,-1,-1,2	0.695	0.715	0.695	0.715	1.000
-1,-1,-1,0,0	0.917	0.928	0.927	0.995	0.993
-1,-1,-1,0,1	0.846	0.866	0.850	0.943	1.000
-1,-1,-1,0,2	0.699	0.704	0.704	0.810	1.000
-1,-1,-1,1,1	0.825	0.812	0.814	0.998	0.996
-1,-1,-1,1,2	0.683	0.689	0.684	0.929	1.000
-1,-1,-1,2,2	0.717	0.733	0.720	0.999	1.000
-1,-1,0,0,0	0.885	0.915	0.995	0.996	0.988
-1,-1,0,0,1	0.811	0.839	0.947	0.949	1.000
-1,-1,0,0,2	0.734	0.734	0.840	0.842	1.000
-1,-1,0,1,1	0.803	0.811	0.927	0.995	0.999
-1,-1,0,1,2	0.707	0.687	0.818	0.911	1.000
-1,-1,0,2,2	0.736	0.743	0.809	0.998	0.997
-1,-1,1,1,1	0.774	0.777	0.985	0.999	0.994
-1,-1,1,1,2	0.696	0.724	0.918	0.902	1.000
-1,-1,1,2,2	0.697	0.689	0.905	0.997	0.999
-1,-1,2,2,2	0.700	0.738	0.998	0.997	0.997
-1,0,0,0,0	0.898	0.993	0.985	0.987	0.995
-1,0,0,0,1	0.833	0.952	0.949	0.951	1.000
-1,0,0,0,2	0.727	0.823	0.837	0.829	1.000
-1,0,0,1,1	0.836	0.942	0.947	0.997	0.999
-1,0,0,1,2	0.748	0.831	0.816	0.928	1.000
-1,0,0,2,2	0.686	0.781	0.774	0.995	0.995
-1,0,1,1,1	0.814	0.919	0.998	0.996	0.997
-1,0,1,1,2	0.718	0.811	0.920	0.928	1.000
-1,0,1,2,2	0.710	0.813	0.911	0.999	0.997
-1,0,2,2,2	0.721	0.835	1.000	0.998	0.998
-1,1,1,1,1	0.788	0.992	0.992	0.994	0.989
-1,1,1,1,2	0.696	0.900	0.906	0.910	1.000
-1,1,1,2,2	0.760	0.930	0.937	0.999	0.999
-1,1,2,2,2	0.741	0.914	0.998	0.998	0.995
-1,2,2,2,2	0.679	0.997	0.994	0.995	0.994
0,0,0,0,0	0.976	0.994	0.985	0.987	0.988
0,0,0,0,1	0.945	0.953	0.943	0.944	1.000
0,0,0,0,2	0.812	0.803	0.833	0.808	1.000
0,0,0,1,1	0.931	0.928	0.936	1.000	0.995
0,0,0,1,2	0.798	0.828	0.820	0.925	1.000
0,0,0,2,2	0.834	0.837	0.823	0.999	0.996
0,0,1,1,1	0.918	0.913	0.997	0.993	0.995
0,0,1,1,2	0.830	0.822	0.919	0.918	1.000
0,0,1,2,2	0.821	0.805	0.900	0.994	1.000
0,0,2,2,2	0.776	0.784	0.999	0.998	0.993
0,1,1,1,1	0.895	0.995	0.995	0.991	0.995
0,1,1,1,2	0.828	0.926	0.930	0.913	1.000
0,1,1,2,2	0.829	0.918	0.910	0.998	0.999
0,1,2,2,2	0.823	0.922	0.996	0.995	0.996
0,2,2,2,2	0.829	0.994	0.992	0.998	0.998
1,1,1,1,1	0.988	0.990	0.995	0.992	0.992
1,1,1,1,2	0.919	0.915	0.899	0.915	1.000
1,1,1,2,2	0.913	0.925	0.909	0.999	0.999
1,1,2,2,2	0.909	0.903	0.997	0.996	0.995
1,2,2,2,2	0.925	0.995	0.997	0.997	0.998
2,2,2,2,2	0.999	0.995	0.996	0.995	0.998

Appendix D. GLOVER and PIRKUL Results under Correlation Structures

Table D.1 Relative Errors by GLOVER and PIRKUL under 2KP Correlation Structure

Correlation Structures	GLOVER Relative Error		PIRKUL Relative Error			Best Legacy Relative Error
	1 Iteration	50 Iteration	1st Solution	Best Solution	# of Comparison	
2,2,2	0.257	0.106	0.257	0.047	42.1	0.272
2,1,1	0.261	0.064	0.261	0.016	44.4	0.364
2,0,0	0.387	0.081	0.387	0.022	43.1	0.685
2,-1,-1	0.384	0.143	0.384	0.035	45.9	0.815
2,-2,-2	0.364	0.181	0.364	0.079	45.8	0.709
1,2,1	0.186	0.083	0.186	0.008	41.0	0.252
1,1,2	0.111	0.045	0.111	0.024	50.6	0.092
1,1,1	0.247	0.105	0.247	0.024	49.7	0.177
1,1,0	0.299	0.112	0.299	0.043	46.4	0.178
1,0,1	0.231	0.089	0.231	0.017	52.2	0.292
1,0,0	0.216	0.061	0.216	0.038	49.9	0.254
1,0,-1	0.328	0.151	0.328	0.081	47.8	0.363
1,-1,0	0.244	0.109	0.244	0.027	52.0	0.331
1,-1,-1	0.344	0.092	0.344	0.054	50.8	0.323
1,-1,-2	0.452	0.081	0.452	0.154	47.7	0.334
1,-2,-1	0.197	0.077	0.197	0.026	52.7	0.223
0,2,0	0.214	0.071	0.214	0.024	42.3	0.621
0,1,1	0.285	0.103	0.285	0.012	51.2	0.315
0,1,0	0.246	0.125	0.246	0.055	49.7	0.318
0,1,-1	0.374	0.198	0.374	0.066	47.2	0.307
0,0,2	0.125	0.074	0.125	0.009	54.4	0.050
0,0,1	0.196	0.058	0.196	0.028	54.1	0.178
0,0,0	0.236	0.097	0.236	0.045	51.7	0.220
0,0,-1	0.275	0.100	0.275	0.068	50.2	0.257
0,0,-2	0.246	0.168	0.246	0.152	48.1	0.218
0,-1,1	0.149	0.058	0.149	0.035	54.5	0.110
0,-1,0	0.239	0.099	0.239	0.038	53.2	0.218
0,-1,-1	0.251	0.066	0.251	0.051	52.3	0.316
0,-2,0	0.290	0.133	0.290	0.058	55.6	0.198
-1,2,-1	0.229	0.085	0.229	0.008	44.5	0.582
-1,1,0	0.250	0.073	0.250	0.022	51.7	0.265
-1,1,-1	0.281	0.105	0.281	0.043	50.8	0.358
-1,1,-2	0.310	0.158	0.310	0.044	47.7	0.290
-1,0,1	0.277	0.083	0.277	0.024	54.6	0.203
-1,0,0	0.292	0.085	0.292	0.026	53.9	0.223
-1,0,-1	0.371	0.131	0.371	0.027	52.3	0.189
-1,-1,2	0.124	0.061	0.124	0.023	57.9	0.111
-1,-1,1	0.203	0.065	0.203	0.027	56.0	0.139
-1,-1,0	0.139	0.029	0.139	0.020	55.4	0.162
-1,-2,1	0.177	0.056	0.177	0.016	58.5	0.064
-2,2,-2	0.677	0.306	0.677	0.131	42.4	0.618
-2,1,-1	0.213	0.055	0.213	0.024	52.4	0.185
-2,0,0	0.313	0.056	0.313	0.041	54.6	0.122
-2,-1,1	0.214	0.072	0.214	0.015	57.9	0.120
-2,-2,2	0.009	0.000	0.009	0.001	60.2	0.005

Table D.2 Relative Errors by GLOVER and PIRKUL under 5KP Correlation Structure

Correlation Structures	GLOVER Relative Error		PIRKUL Relative Error			Best Legacy Relative Error
	1 Iteration	50 Iteration	1st Solution	Best Solution	# of Comparison	
-2,-2,-2,-2,-2	0.189	0.060	0.189	0.068	56.4	0.123
-2,-2,-2,-2,-1	0.186	0.053	0.186	0.060	55.7	0.179
-2,-2,-2,-2,0	0.311	0.080	0.311	0.073	54.3	0.371
-2,-2,-2,-2,1	0.414	0.104	0.414	0.107	53.4	0.623
-2,-2,-2,-2,2	0.548	0.214	0.548	0.078	51.1	0.874
-2,-2,-2,-1,-1	0.347	0.155	0.347	0.103	54.5	0.356
-2,-2,-2,-1,0	0.343	0.100	0.343	0.081	54.0	0.443
-2,-2,-2,-1,1	0.404	0.068	0.404	0.082	52.7	0.825
-2,-2,-2,-1,2	0.360	0.118	0.360	0.093	50.2	0.638
-2,-2,-2,0,0	0.304	0.131	0.304	0.048	52.5	0.573
-2,-2,-2,0,1	0.350	0.094	0.350	0.047	51.3	0.713
-2,-2,-2,0,2	0.526	0.180	0.526	0.120	49.9	1.171
-2,-2,-2,1,1	0.344	0.102	0.344	0.091	49.7	0.888
-2,-2,-2,1,2	0.633	0.247	0.633	0.167	48.2	1.744
-2,-2,-2,2,2	0.492	0.211	0.492	0.144	46.6	1.678
-2,-2,-1,-1,-1	0.428	0.123	0.428	0.053	53.5	0.307
-2,-2,-1,-1,0	0.376	0.137	0.376	0.108	52.4	0.382
-2,-2,-1,-1,1	0.361	0.081	0.361	0.054	51.6	0.648
-2,-2,-1,-1,2	0.618	0.141	0.618	0.146	49.5	1.162
-2,-2,-1,0,0	0.359	0.105	0.359	0.063	51.4	0.529
-2,-2,-1,0,1	0.408	0.129	0.408	0.096	50.2	0.862
-2,-2,-1,0,2	0.653	0.228	0.653	0.112	48.5	1.231
-2,-2,-1,1,1	0.459	0.163	0.459	0.088	49.2	1.047
-2,-2,-1,1,2	0.455	0.193	0.455	0.100	48.1	1.639
-2,-2,-1,2,2	0.763	0.396	0.763	0.202	47.0	1.658
-2,-2,0,0,0	0.377	0.118	0.377	0.077	49.1	0.442
-2,-2,0,0,1	0.447	0.167	0.447	0.115	49.1	0.865
-2,-2,0,0,2	0.436	0.143	0.436	0.069	48.3	1.820
-2,-2,0,1,1	0.530	0.196	0.530	0.098	47.7	0.847
-2,-2,0,1,2	0.597	0.259	0.597	0.098	46.9	2.068
-2,-2,0,2,2	0.770	0.207	0.770	0.110	45.9	1.779
-2,-2,1,1,1	0.721	0.230	0.721	0.139	46.7	1.069
-2,-2,1,1,2	0.703	0.343	0.703	0.127	45.5	1.478
-2,-2,1,2,2	0.748	0.308	0.748	0.148	44.6	1.668
-2,-2,2,2,2	0.906	0.349	0.906	0.204	43.4	1.258
-2,-1,-1,-1,-1	0.455	0.093	0.455	0.097	51.9	0.317
-2,-1,-1,-1,0	0.397	0.078	0.397	0.093	51.1	0.540
-2,-1,-1,-1,1	0.516	0.215	0.516	0.103	49.6	0.781
-2,-1,-1,-1,2	0.402	0.099	0.402	0.089	48.8	1.221
-2,-1,-1,0,0	0.487	0.135	0.487	0.104	50.0	0.737
-2,-1,-1,0,1	0.510	0.133	0.510	0.111	49.0	0.877
-2,-1,-1,0,2	0.398	0.161	0.398	0.115	47.7	1.106
-2,-1,-1,1,1	0.382	0.143	0.382	0.050	47.7	0.940
-2,-1,-1,1,2	0.517	0.196	0.517	0.092	47.2	1.581
-2,-1,-1,2,2	0.541	0.185	0.541	0.131	45.8	1.633
-2,-1,0,0,0	0.494	0.196	0.494	0.093	48.6	0.677
-2,-1,0,0,1	0.513	0.200	0.513	0.109	47.7	0.851
-2,-1,0,0,2	0.682	0.184	0.682	0.130	46.3	1.326
-2,-1,0,1,1	0.560	0.181	0.560	0.111	46.8	0.978
-2,-1,0,1,2	0.484	0.268	0.484	0.112	46.3	1.480
-2,-1,0,2,2	0.565	0.297	0.565	0.118	44.3	1.698
-2,-1,1,1,1	0.625	0.233	0.625	0.123	45.7	1.114
-2,-1,1,1,2	0.729	0.210	0.729	0.170	45.2	1.743
-2,-1,1,2,2	0.753	0.234	0.753	0.102	43.9	1.739
-2,-1,2,2,2	0.983	0.330	0.983	0.186	42.9	1.365
-2,0,0,0,0	0.641	0.237	0.641	0.156	47.0	0.502
-2,0,0,0,1	0.814	0.220	0.814	0.161	46.1	0.913
-2,0,0,0,2	0.558	0.151	0.558	0.099	44.7	1.710
-2,0,0,1,1	0.619	0.220	0.619	0.156	45.4	0.935
-2,0,0,1,2	0.604	0.212	0.604	0.105	44.4	1.426
-2,0,0,2,2	0.671	0.257	0.671	0.133	44.2	1.562
-2,0,1,1,1	0.642	0.227	0.642	0.131	44.1	0.848

Correlation Structures	GLOVER Relative Error		PIRKUL Relative Error			Best Legacy Relative Error
	1 Iteration	50 Iteration	1st Solution	Best Solution	# of Comparison	
-2,0,1,1,2	0.766	0.390	0.766	0.143	43.8	1.494
-2,0,1,2,2	0.821	0.277	0.821	0.167	43.2	1.684
-2,0,2,2,2	0.835	0.219	0.835	0.166	42.1	1.449
-2,1,1,1,1	0.582	0.231	0.582	0.158	43.2	0.573
-2,1,1,1,2	0.754	0.339	0.754	0.162	43.0	1.633
-2,1,1,2,2	0.756	0.345	0.756	0.182	42.5	1.654
-2,1,2,2,2	1.029	0.427	1.029	0.214	41.6	1.434
-2,2,2,2,2	0.778	0.311	0.778	0.197	39.8	0.945
-1,-1,-1,-1,-1	0.443	0.148	0.443	0.157	50.2	0.340
-1,-1,-1,-1,0	0.564	0.172	0.564	0.140	49.6	0.451
-1,-1,-1,-1,1	0.567	0.103	0.567	0.085	48.2	0.869
-1,-1,-1,-1,2	0.558	0.191	0.558	0.097	47.1	1.025
-1,-1,-1,0,0	0.587	0.201	0.587	0.172	48.6	0.565
-1,-1,-1,0,1	0.689	0.281	0.689	0.196	47.8	0.664
-1,-1,-1,0,2	0.680	0.184	0.680	0.120	47.0	1.396
-1,-1,-1,1,1	0.526	0.261	0.526	0.137	46.7	1.212
-1,-1,-1,1,2	0.629	0.220	0.629	0.139	45.5	1.672
-1,-1,-1,2,2	0.603	0.184	0.603	0.114	44.6	1.455
-1,-1,0,0,0	0.364	0.116	0.364	0.117	47.4	0.725
-1,-1,0,0,1	0.592	0.189	0.592	0.149	46.7	0.852
-1,-1,0,0,2	0.555	0.175	0.555	0.149	45.9	1.600
-1,-1,0,1,1	0.518	0.160	0.518	0.126	45.8	0.968
-1,-1,0,1,2	0.569	0.192	0.569	0.129	45.2	1.503
-1,-1,0,2,2	0.717	0.183	0.717	0.165	44.5	1.681
-1,-1,1,1,1	0.710	0.252	0.710	0.114	44.2	1.079
-1,-1,1,1,2	0.669	0.279	0.669	0.119	44.4	1.736
-1,-1,1,2,2	0.762	0.275	0.762	0.150	42.7	1.697
-1,-1,2,2,2	0.616	0.266	0.616	0.151	42.3	1.653
-1,0,0,0,0	0.561	0.228	0.561	0.132	45.9	0.690
-1,0,0,0,1	0.515	0.190	0.515	0.101	44.9	0.942
-1,0,0,0,2	0.817	0.331	0.817	0.116	44.9	1.386
-1,0,0,1,1	0.686	0.331	0.686	0.157	44.2	1.144
-1,0,0,1,2	0.784	0.210	0.784	0.136	43.8	1.778
-1,0,0,2,2	0.495	0.287	0.495	0.173	42.9	1.759
-1,0,1,1,1	0.676	0.294	0.676	0.184	43.1	0.867
-1,0,1,1,2	0.566	0.207	0.566	0.102	43.2	1.909
-1,0,1,2,2	0.920	0.341	0.920	0.194	42.7	1.675
-1,0,2,2,2	0.831	0.248	0.831	0.135	40.9	1.891
-1,1,1,1,1	0.628	0.309	0.628	0.201	41.9	0.940
-1,1,1,1,2	0.690	0.358	0.690	0.138	41.9	1.444
-1,1,1,2,2	0.944	0.405	0.944	0.153	41.9	1.561
-1,1,2,2,2	0.648	0.288	0.648	0.177	41.1	1.499
-1,2,2,2,2	0.609	0.229	0.609	0.158	39.0	1.041
0,0,0,0,0	0.578	0.340	0.578	0.131	44.0	0.542
0,0,0,0,1	0.595	0.261	0.595	0.191	43.6	0.827
0,0,0,0,2	0.631	0.318	0.631	0.148	42.2	1.409
0,0,0,1,1	0.767	0.443	0.767	0.158	42.2	1.119
0,0,0,1,2	0.687	0.288	0.687	0.211	42.3	1.833
0,0,0,2,2	0.840	0.268	0.840	0.111	41.6	1.797
0,0,1,1,1	0.745	0.344	0.745	0.183	41.7	0.771
0,0,1,1,2	0.820	0.397	0.820	0.220	41.6	1.539
0,0,1,2,2	0.763	0.429	0.763	0.177	41.6	1.537
0,0,2,2,2	0.743	0.286	0.743	0.156	40.3	1.675
0,1,1,1,1	0.654	0.387	0.654	0.193	40.5	1.007
0,1,1,1,2	1.084	0.347	1.084	0.169	41.0	1.718
0,1,1,2,2	0.916	0.419	0.916	0.173	40.0	1.739
0,1,2,2,2	0.596	0.263	0.596	0.176	40.2	1.341
0,2,2,2,2	0.523	0.256	0.523	0.188	39.5	1.231
1,1,1,1,1	0.623	0.379	0.623	0.193	39.1	0.668
1,1,1,1,2	0.601	0.305	0.601	0.180	39.6	1.261
1,1,1,2,2	0.652	0.336	0.652	0.186	39.2	1.464
1,1,2,2,2	0.637	0.408	0.637	0.154	39.3	1.475
1,2,2,2,2	0.655	0.316	0.655	0.209	38.3	1.269
2,2,2,2,2	0.641	0.362	0.641	0.191	37.9	0.485

Appendix E. Results by Correlation Structures for New Heuristics

Table E.1 Number of Times Equal to Best by TYPE under 2KP Correlation Structure

	TYPE	TOYODA	S-T	L-M M1	FOX	KOCHEN	Total Probs
2,2,2	**5**	0	0	0	5	5	20
2,1,1	**13**	1	0	1	13	1	20
2,0,0	**11**	0	0	1	11	3	20
2,-1,-1	**14**	0	0	0	12	5	20
2,-2,-2	**14**	1	1	0	9	9	20
1,2,1	**14**	1	2	1	14	0	20
1,1,2	**1**	0	0	0	6	1	20
1,1,1	**12**	0	5	0	4	8	40
1,1,0	**12**	0	6	0	0	6	20
1,0,1	**4**	0	0	0	9	4	20
1,0,0	**24**	1	5	0	1	22	40
1,0,-1	**17**	0	7	0	0	10	20
1,-1,0	**6**	1	1	0	6	8	20
1,-1,-1	**23**	0	9	0	3	18	40
1,-1,-2	**15**	0	5	0	0	12	20
1,-2,-1	**5**	0	1	0	2	7	20
0,2,0	**14**	0	1	0	14	1	20
0,1,1	**3**	0	1	0	9	2	20
0,1,0	**23**	0	5	1	1	21	40
0,1,-1	**17**	0	7	0	0	11	20
0,0,2	**0**	2	0	1	1	0	20
0,0,1	**7**	2	2	2	4	8	40
0,0,0	**12**	1	6	1	1	10	40
0,0,-1	**18**	1	11	0	0	9	40
0,0,-2	**10**	0	8	1	0	3	20
0,-1,1	**4**	3	0	0	4	4	20
0,-1,0	**15**	0	3	4	3	13	40
0,-1,-1	**7**	1	3	4	1	4	20
0,-2,0	**3**	1	0	1	6	3	20
-1,2,-1	**15**	1	1	1	14	2	20
-1,1,0	**7**	0	0	2	5	7	20
-1,1,-1	**23**	0	7	1	3	23	40
-1,1,-2	**12**	1	5	2	0	11	20
-1,0,1	**2**	1	0	0	7	2	20
-1,0,0	**17**	1	4	3	2	15	40
-1,0,-1	**8**	1	2	2	0	6	20
-1,-1,2	**0**	1	0	3	1	0	20
-1,-1,1	**3**	2	1	2	6	4	40
-1,-1,0	**3**	1	2	3	0	3	20
-1,-2,1	**5**	0	1	0	3	4	20
-2,2,-2	**14**	0	1	2	11	4	20
-2,1,-1	**6**	0	0	1	3	9	20
-2,0,0	**5**	0	1	3	4	4	20
-2,-1,1	**3**	0	0	0	5	3	20
-2,-2,2	**0**	0	0	0	0	0	20

Table E.2 Number of Times Equal to Best by TYPE under 5KP Correlation Structure

Correlation	**TYPE**	TOYODA	S-T	L-M M1	FOX	KOCHEN	Total Probs
-2,-2,-2,-2,-2	**2**	0	0	6	8	2	30
-2,-2,-2,-2,-1	**6**	1	1	5	6	5	30
-2,-2,-2,-2,0	**5**	2	1	7	7	6	30
-2,-2,-2,-2,1	**3**	0	3	6	6	3	30
-2,-2,-2,-2,2	**15**	2	0	5	9	10	30
-2,-2,-2,-1,-1	**9**	0	1	7	4	9	30
-2,-2,-2,-1,0	**5**	1	0	13	2	5	30
-2,-2,-2,-1,1	**8**	0	0	14	3	8	30
-2,-2,-2,-1,2	**14**	1	0	7	10	6	30
-2,-2,-2,0,0	**12**	2	3	9	0	11	30
-2,-2,-2,0,1	**17**	0	3	3	1	15	30
-2,-2,-2,0,2	**14**	2	0	5	8	12	30
-2,-2,-2,1,1	**19**	0	4	8	0	15	30
-2,-2,-2,1,2	**9**	0	1	8	5	14	30
-2,-2,-2,2,2	**17**	1	2	3	3	19	30
-2,-2,-1,-1,-1	**14**	1	1	4	3	15	30
-2,-2,-1,-1,0	**10**	2	0	8	2	10	30
-2,-2,-1,-1,1	**8**	0	1	12	4	10	30
-2,-2,-1,-1,2	**17**	0	0	5	8	13	30
-2,-2,-1,0,0	**18**	0	2	4	1	17	30
-2,-2,-1,0,1	**12**	0	2	12	0	12	30
-2,-2,-1,0,2	**19**	0	0	4	8	17	30
-2,-2,-1,1,1	**15**	0	3	10	0	14	30
-2,-2,-1,1,2	**12**	1	1	5	5	15	30
-2,-2,-1,2,2	**22**	0	0	3	4	23	30
-2,-2,0,0,0	**13**	1	1	7	4	13	30
-2,-2,0,0,1	**19**	1	3	4	1	17	30
-2,-2,0,0,2	**21**	0	1	2	5	22	30
-2,-2,0,1,1	**20**	1	1	6	0	20	30
-2,-2,0,1,2	**14**	0	1	9	3	15	30
-2,-2,0,2,2	**18**	1	0	5	3	21	30
-2,-2,1,1,1	**22**	0	2	5	0	21	30
-2,-2,1,1,2	**12**	1	0	4	1	23	30
-2,-2,1,2,2	**18**	0	1	0	3	25	30
-2,-2,2,2,2	**20**	1	1	3	3	22	30
-2,-1,-1,-1,-1	**16**	0	1	7	1	15	30
-2,-1,-1,-1,0	**10**	0	1	14	2	10	30
-2,-1,-1,-1,1	**11**	1	2	10	3	9	30
-2,-1,-1,-1,2	**17**	1	1	4	7	13	30
-2,-1,-1,0,0	**15**	1	0	8	0	16	30
-2,-1,-1,0,1	**21**	0	3	7	0	19	30
-2,-1,-1,0,2	**18**	3	1	5	9	10	30
-2,-1,-1,1,1	**22**	0	4	4	0	18	30
-2,-1,-1,1,2	**12**	0	1	5	4	18	30
-2,-1,-1,2,2	**15**	1	2	3	3	16	30
-2,-1,0,0,0	**21**	1	2	4	0	19	30
-2,-1,0,0,1	**15**	0	0	11	1	15	30
-2,-1,0,0,2	**17**	0	0	6	6	17	30
-2,-1,0,1,1	**20**	0	0	7	0	20	30
-2,-1,0,1,2	**17**	0	0	4	5	21	30
-2,-1,0,2,2	**15**	2	0	6	1	21	30
-2,-1,1,1,1	**23**	0	2	3	0	21	30
-2,-1,1,1,2	**16**	0	1	5	2	22	30
-2,-1,1,2,2	**18**	0	0	3	1	25	30
-2,-1,2,2,2	**18**	0	2	4	3	21	30
-2,0,0,0,0	**18**	0	2	2	2	18	30
-2,0,0,0,1	**19**	0	1	7	0	18	30
-2,0,0,0,2	**17**	0	1	3	4	19	30
-2,0,0,1,1	**25**	1	2	1	0	23	30
-2,0,0,1,2	**14**	2	1	6	2	19	30
-2,0,0,2,2	**21**	1	0	3	2	22	30
-2,0,1,1,1	**25**	0	1	2	0	24	30
-2,0,1,1,2	**17**	0	1	3	2	24	30
-2,0,1,2,2	**18**	0	1	5	0	24	30

Correlation	**TYPE**	TOYODA	S-T	L-M M1	FOX	KOCHEN	Total Probs
-2,0,2,2,2	**20**	0	0	2	2	25	30
-2,1,1,1,1	**21**	5	2	1	1	19	30
-2,1,1,1,2	**14**	0	0	2	1	27	30
-2,1,1,2,2	**19**	0	0	2	0	25	30
-2,1,2,2,2	**22**	0	1	2	1	26	30
-2,2,2,2,2	**17**	3	0	0	0	25	30
-1,-1,-1,-1,-1	**8**	1	1	2	1	9	30
-1,-1,-1,-1,0	**9**	3	1	10	1	8	30
-1,-1,-1,-1,1	**12**	1	3	7	3	12	30
-1,-1,-1,-1,2	**16**	0	3	5	7	10	30
-1,-1,-1,0,0	**18**	1	5	4	0	15	30
-1,-1,-1,0,1	**16**	2	3	4	0	14	30
-1,-1,-1,0,2	**18**	0	4	6	7	10	30
-1,-1,-1,1,1	**21**	1	4	4	0	18	30
-1,-1,-1,1,2	**12**	2	1	4	2	17	30
-1,-1,-1,2,2	**17**	1	1	4	2	17	30
-1,-1,0,0,0	**17**	0	0	3	0	18	30
-1,-1,0,0,1	**15**	1	1	3	3	16	30
-1,-1,0,0,2	**17**	0	5	5	3	15	30
-1,-1,0,1,1	**21**	0	6	4	0	18	30
-1,-1,0,1,2	**17**	0	3	3	5	15	30
-1,-1,0,2,2	**21**	1	3	3	4	16	30
-1,-1,1,1,1	**19**	2	1	3	0	22	30
-1,-1,1,1,2	**14**	2	2	7	1	18	30
-1,-1,1,2,2	**19**	0	2	3	1	23	30
-1,-1,2,2,2	**22**	2	4	0	2	19	30
-1,0,0,0,0	**22**	2	2	1	0	20	30
-1,0,0,0,1	**21**	1	2	5	0	19	30
-1,0,0,0,2	**18**	0	2	4	3	18	30
-1,0,0,1,1	**26**	0	2	1	0	25	30
-1,0,0,1,2	**16**	0	4	5	1	17	30
-1,0,0,2,2	**21**	0	2	2	0	24	30
-1,0,1,1,1	**28**	0	3	0	0	27	30
-1,0,1,1,2	**17**	0	4	3	0	22	30
-1,0,1,2,2	**20**	0	2	5	0	21	30
-1,0,2,2,2	**25**	0	4	2	1	23	30
-1,1,1,1,1	**25**	2	4	0	1	22	30
-1,1,1,1,2	**19**	0	4	6	1	18	30
-1,1,1,2,2	**22**	0	4	3	0	23	30
-1,1,2,2,2	**23**	0	3	0	0	27	30
-1,2,2,2,2	**25**	1	4	1	2	22	30
0,0,0,0,0	**14**	3	5	1	0	12	30
0,0,0,0,1	**13**	1	5	3	1	13	30
0,0,0,0,2	**14**	1	7	2	3	12	30
0,0,0,1,1	**17**	2	6	2	0	17	30
0,0,0,1,2	**14**	0	8	2	2	15	30
0,0,0,2,2	**19**	1	2	3	0	20	30
0,0,1,1,1	**23**	1	6	1	0	21	30
0,0,1,1,2	**16**	1	5	2	2	17	30
0,0,1,2,2	**16**	1	5	5	0	16	30
0,0,2,2,2	**22**	0	5	1	1	20	30
0,1,1,1,1	**27**	0	4	0	0	24	30
0,1,1,1,2	**17**	0	2	2	0	24	30
0,1,1,2,2	**22**	0	2	2	0	23	30
0,1,2,2,2	**25**	0	5	0	0	25	30
0,2,2,2,2	**24**	0	4	0	0	25	30
1,1,1,1,1	**21**	1	3	0	1	19	30
1,1,1,1,2	**9**	4	7	4	0	12	30
1,1,1,2,2	**19**	1	5	3	0	19	30
1,1,2,2,2	**22**	0	6	2	1	19	30
1,2,2,2,2	**24**	1	4	0	1	23	30
2,2,2,2,2	**21**	0	2	0	1	20	30

Table E.3 Sign Test for NG V1 based on Constraint Slackness

Prob Type	Constraint Slackness	NG V1 Heuristic		vs. Legacy Heuristic		Pr(X ≥ U)	Reject Region (α = 0.1)
2KP	1, 1	NG V1	216	TOYODA	51	0.000	∗
			229	S-T	39	0.000	∗
			247	L-MM1	26	0.000	∗
			219	FOX	44	0.000	∗
			125	KOCHEN	99	0.047	∗
	1, 2	NG V1	220	TOYODA	25	0.000	∗
			100	S-T	53	0.000	∗
			246	L-MM1	13	0.000	∗
			184	FOX	20	0.000	∗
			112	KOCHEN	48	0.000	∗
	2, 1	NG V1	215	TOYODA	24	0.000	∗
			98	S-T	58	0.001	∗
			250	L-MM1	9	0.000	∗
			169	FOX	21	0.000	∗
			103	KOCHEN	53	0.000	∗
	2, 2	NG V1	205	TOYODA	48	0.000	∗
			211	S-T	45	0.000	∗
			217	L-MM1	37	0.000	∗
			160	FOX	59	0.000	∗
			123	KOCHEN	78	0.001	∗
5KP	1,1,1,1,1	NG V1	587	TOYODA	39	0.000	∗
			622	S-T	7	0.000	∗
			533	L-MM1	90	0.000	∗
			561	FOX	69	0.000	∗
			273	KOCHEN	287	0.263	
	1,1,1,1,2	NG V1	559	TOYODA	62	0.000	∗
			595	S-T	25	0.000	∗
			542	L-MM1	77	0.000	∗
			597	FOX	32	0.000	∗
			273	KOCHEN	307	0.073	
	1,1,1,2,2	NG V1	523	TOYODA	90	0.000	∗
			557	S-T	62	0.000	∗
			531	L-MM1	67	0.000	∗
			603	FOX	23	0.000	∗
			311	KOCHEN	267	0.037	∗
	1,1,2,2,2	NG V1	529	TOYODA	85	0.000	∗
			483	S-T	123	0.000	∗
			483	L-MM1	85	0.000	∗
			617	FOX	8	0.000	∗
			335	KOCHEN	240	0.000	∗
	1,2,2,2,2	NG V1	575	TOYODA	37	0.000	∗
			360	S-T	115	0.000	∗
			471	L-MM1	91	0.000	∗
			597	FOX	5	0.000	∗
			360	KOCHEN	160	0.000	∗
	2,2,2,2,2	NG V1	578	TOYODA	36	0.000	∗
			591	S-T	27	0.000	∗
			549	L-MM1	46	0.000	∗
			524	FOX	87	0.000	∗
			297	KOCHEN	272	0.157	

U ~ the number times best heuristic is better than compared heuristic
∗ indicates NG V1 as the best

Table E.4 Number of Times Best by NG V1 and Each Heuristic under 2KP Correlation Structure

	NG V1	TOYODA	S-T	L-M M1	FOX	KOCHEN	Sign Test
2,2,2	**10**	0	0	0	2	1	∗
2,1,1	**2**	1	0	1	12	0	
2,0,0	**2**	0	0	1	10	1	
2,-1,-1	**4**	0	0	0	8	1	
2,-2,-2	**5**	1	0	0	7	1	∗
1,2,1	**2**	1	0	1	12	0	
1,1,2	**0**	0	0	0	4	1	
1,1,1	**5**	0	4	0	1	6	
1,1,0	**5**	0	3	0	0	5	
1,0,1	**7**	0	0	0	2	2	∗
1,0,0	**20**	0	2	0	0	6	∗
1,0,-1	**2**	0	5	0	0	9	
1,-1,0	**5**	1	1	0	5	2	
1,-1,-1	**20**	0	3	0	0	5	∗
1,-1,-2	**6**	0	0	0	0	8	
1,-2,-1	**4**	0	1	0	2	0	∗
0,2,0	**4**	0	0	0	11	0	
0,1,1	**4**	0	1	0	3	0	∗
0,1,0	**18**	0	1	1	0	7	∗
0,1,-1	**4**	0	4	0	0	7	
0,0,2	**0**	2	0	1	0	0	
0,0,1	**8**	2	1	1	2	4	
0,0,0	**7**	1	2	1	1	7	
0,0,-1	**12**	1	2	0	0	9	
0,0,-2	**5**	0	1	1	0	3	
0,-1,1	**1**	3	0	0	1	3	
0,-1,0	**13**	0	2	2	2	5	∗
0,-1,-1	**6**	1	1	3	0	1	
0,-2,0	**5**	1	0	0	1	1	∗
-1,2,-1	**4**	1	1	1	11	0	
-1,1,0	**5**	0	0	2	3	1	∗
-1,1,-1	**12**	0	2	0	1	10	∗
-1,1,-2	**9**	1	2	1	0	3	∗
-1,0,1	**3**	1	0	0	3	0	
-1,0,0	**11**	1	2	0	0	7	
-1,0,-1	**6**	0	2	1	0	2	
-1,-1,2	**0**	1	0	3	1	0	
-1,-1,1	**4**	2	1	2	1	3	
-1,-1,0	**3**	1	1	2	0	2	
-1,-2,1	**2**	0	1	0	1	3	
-2,2,-2	**3**	0	0	2	9	2	
-2,1,-1	**3**	0	0	0	3	3	∗
-2,0,0	**3**	0	1	1	0	2	∗
-2,-1,1	**4**	0	0	0	2	0	∗
-2,-2,2	**0**	0	0	0	0	0	

(∗ indicates NG V1 as the best, Reject Region: $\alpha = 0.1$)

Table E. 5 Number of Times Best by NG V1 and Each Heuristic under 5KP Correlation Structure

Correlation	NG V1	TOYODA	S-T	L-M M1	FOX	KOCHEN	Sign Test
-2,-2,-2,-2,-2	**5**	0	0	2	6	0	*
-2,-2,-2,-2,-1	**6**	1	1	3	3	1	
-2,-2,-2,-2,0	**8**	1	0	3	5	5	*
-2,-2,-2,-2,1	**7**	0	2	4	5	0	
-2,-2,-2,-2,2	**5**	1	0	3	9	5	
-2,-2,-2,-1,-1	**8**	0	1	3	3	3	
-2,-2,-2,-1,0	**12**	0	0	4	2	0	*
-2,-2,-2,-1,1	**10**	0	0	6	2	2	*
-2,-2,-2,-1,2	**8**	1	0	5	10	1	*
-2,-2,-2,0,0	**8**	1	1	2	0	9	
-2,-2,-2,0,1	**11**	0	3	2	1	3	
-2,-2,-2,0,2	**10**	1	0	1	7	4	*
-2,-2,-2,1,1	**10**	0	1	3	0	11	
-2,-2,-2,1,2	**9**	0	1	7	3	8	
-2,-2,-2,2,2	**6**	0	1	2	2	16	
-2,-2,-1,-1,-1	**8**	0	0	3	1	6	
-2,-2,-1,-1,0	**11**	2	0	2	1	2	*
-2,-2,-1,-1,1	**11**	0	0	4	2	7	*
-2,-2,-1,-1,2	**6**	0	0	3	8	9	
-2,-2,-1,0,0	**11**	0	1	2	1	8	
-2,-2,-1,0,1	**15**	0	1	5	0	3	*
-2,-2,-1,0,2	**11**	0	0	2	8	3	*
-2,-2,-1,1,1	**12**	0	1	3	0	8	
-2,-2,-1,1,2	**15**	1	0	0	5	6	*
-2,-2,-1,2,2	**3**	0	0	2	4	18	
-2,-2,0,0,0	**5**	0	1	3	3	8	
-2,-2,0,0,1	**12**	1	1	1	0	6	*
-2,-2,0,0,2	**15**	0	0	0	2	9	*
-2,-2,0,1,1	**12**	1	1	1	0	11	*
-2,-2,0,1,2	**21**	0	1	3	0	4	*
-2,-2,0,2,2	**8**	1	0	3	2	15	
-2,-2,1,1,1	**7**	0	2	2	0	16	
-2,-2,1,1,2	**15**	0	0	2	0	10	*
-2,-2,1,2,2	**8**	0	0	0	3	17	
-2,-2,2,2,2	**7**	1	0	1	3	16	
-2,-1,-1,-1,-1	**11**	0	0	5	0	7	*
-2,-1,-1,-1,0	**11**	0	1	3	1	5	*
-2,-1,-1,-1,1	**12**	1	1	5	1	1	*
-2,-1,-1,-1,2	**8**	1	0	2	7	4	*
-2,-1,-1,0,0	**12**	0	0	3	0	9	
-2,-1,-1,0,1	**12**	0	2	2	0	11	
-2,-1,-1,0,2	**9**	1	0	3	9	7	*
-2,-1,-1,1,1	**8**	0	1	2	0	12	
-2,-1,-1,1,2	**15**	0	0	2	3	6	*
-2,-1,-1,2,2	**8**	1	1	1	2	13	
-2,-1,0,0,0	**10**	1	1	3	0	11	
-2,-1,0,0,1	**10**	0	0	3	0	9	
-2,-1,0,0,2	**12**	0	0	4	5	6	*
-2,-1,0,1,1	**13**	0	0	1	0	10	
-2,-1,0,1,2	**17**	0	0	1	3	7	*
-2,-1,0,2,2	**8**	0	0	2	1	14	
-2,-1,1,1,1	**11**	0	0	1	0	14	
-2,-1,1,1,2	**15**	0	1	3	0	7	*
-2,-1,1,2,2	**15**	0	0	1	0	13	
-2,-1,2,2,2	**4**	0	2	1	3	18	
-2,0,0,0,0	**8**	0	1	1	0	12	
-2,0,0,0,1	**15**	0	0	1	0	9	
-2,0,0,0,2	**12**	0	0	0	2	9	
-2,0,0,1,1	**9**	0	1	1	0	14	
-2,0,0,1,2	**18**	1	0	2	0	7	*
-2,0,0,2,2	**11**	0	0	2	2	14	
-2,0,1,1,1	**12**	0	0	1	0	11	
-2,0,1,1,2	**17**	0	1	1	1	9	*

Correlation	**NG V1**	TOYODA	S-T	L-M M1	FOX	KOCHEN	Sign Test
-2,0,1,2,2	**11**	0	0	5	0	14	
-2,0,2,2,2	**8**	0	0	1	2	18	
-2,1,1,1,1	**5**	5	1	1	1	14	
-2,1,1,1,2	**16**	0	0	1	0	12	
-2,1,1,2,2	**11**	0	0	1	0	15	
-2,1,2,2,2	**9**	0	0	2	0	15	
-2,2,2,2,2	**11**	1	0	0	0	15	
-1,-1,-1,-1,-1	**8**	1	1	1	1	7	
-1,-1,-1,-1,0	**7**	1	1	4	1	6	*
-1,-1,-1,-1,1	**11**	1	2	3	2	4	*
-1,-1,-1,-1,2	**10**	0	2	3	7	3	*
-1,-1,-1,0,0	**6**	1	2	2	0	12	
-1,-1,-1,0,1	**15**	0	2	1	0	4	*
-1,-1,-1,0,2	**13**	0	4	1	4	3	*
-1,-1,-1,1,1	**15**	0	3	0	0	11	*
-1,-1,-1,1,2	**15**	1	0	3	1	6	*
-1,-1,-1,2,2	**7**	0	0	2	2	13	
-1,-1,0,0,0	**10**	0	0	1	0	9	
-1,-1,0,0,1	**17**	1	1	2	0	5	*
-1,-1,0,0,2	**9**	0	2	3	2	9	
-1,-1,0,1,1	**14**	0	2	2	0	9	*
-1,-1,0,1,2	**13**	0	2	2	4	4	*
-1,-1,0,2,2	**5**	1	3	2	4	11	
-1,-1,1,1,1	**10**	1	0	2	0	14	
-1,-1,1,1,2	**18**	0	0	3	0	7	*
-1,-1,1,2,2	**18**	0	1	1	0	9	*
-1,-1,2,2,2	**3**	2	1	0	2	15	
-1,0,0,0,0	**9**	2	1	1	0	12	
-1,0,0,0,1	**14**	0	0	1	0	13	
-1,0,0,0,2	**10**	0	1	3	2	11	
-1,0,0,1,1	**12**	0	0	1	0	14	
-1,0,0,1,2	**19**	0	1	1	0	6	*
-1,0,0,2,2	**13**	0	1	2	0	11	
-1,0,1,1,1	**8**	0	1	0	0	21	
-1,0,1,1,2	**21**	0	0	1	0	7	*
-1,0,1,2,2	**9**	0	2	4	0	13	
-1,0,2,2,2	**9**	0	2	0	1	16	
-1,1,1,1,1	**11**	1	0	0	0	15	
-1,1,1,1,2	**16**	0	0	1	0	9	*
-1,1,1,2,2	**11**	0	1	1	0	13	
-1,1,2,2,2	**10**	0	1	0	0	17	
-1,2,2,2,2	**8**	0	1	0	1	18	
0,0,0,0,0	**12**	2	3	0	0	8	
0,0,0,0,1	**13**	1	3	0	0	7	
0,0,0,0,2	**13**	1	1	2	3	5	*
0,0,0,1,1	**15**	2	5	0	0	8	
0,0,0,1,2	**18**	0	3	1	2	4	*
0,0,0,2,2	**4**	1	1	2	0	15	
0,0,1,1,1	**8**	1	3	0	0	13	
0,0,1,1,2	**15**	0	3	0	0	5	*
0,0,1,2,2	**7**	1	3	5	0	10	
0,0,2,2,2	**7**	0	2	1	1	15	
0,1,1,1,1	**10**	0	1	0	0	18	
0,1,1,1,2	**19**	0	0	0	0	10	*
0,1,1,2,2	**12**	0	1	1	0	13	
0,1,2,2,2	**10**	0	1	0	0	15	
0,2,2,2,2	**6**	0	0	0	0	20	
1,1,1,1,1	**14**	0	0	0	0	12	
1,1,1,1,2	**16**	3	3	1	0	3	
1,1,1,2,2	**12**	0	1	1	0	11	
1,1,2,2,2	**8**	0	2	0	0	15	
1,2,2,2,2	**11**	0	1	0	0	17	
2,2,2,2,2	**11**	0	0	0	0	14	

(∗ indicates NG V1 as the best, Reject Region: $\alpha = 0.2$)

Table E.6 Sign Test for NG V2 based on Constraint Slackness

Prob Type	Constraint Slackness	NG V2 Heuristic		vs. Legacy Heuristic		Pr(X ≥ U)	Reject Region (α = 0.1)
2KP	1, 1	NG V2	221	TOYODA	19	0.000	∗
			230	S-T	17	0.000	∗
			264	L-MM1	7	0.000	∗
			240	FOX	20	0.000	∗
			130	KOCHEN	40	0.000	∗
	1, 2	NG V2	217	TOYODA	27	0.000	∗
			98	S-T	71	0.023	∗
			250	L-MM1	11	0.000	∗
			179	FOX	5	0.000	∗
			107	KOCHEN	65	0.001	∗
	2, 1	NG V2	214	TOYODA	26	0.000	∗
			101	S-T	72	0.017	∗
			249	L-MM1	11	0.000	∗
			167	FOX	5	0.000	∗
			103	KOCHEN	55	0.000	∗
	2, 2	NG V2	208	TOYODA	15	0.000	∗
			212	S-T	14	0.000	∗
			242	L-MM1	14	0.000	∗
			186	FOX	33	0.000	∗
			109	KOCHEN	10	0.000	∗
5KP	1,1,1,1,1	NG V2	618	TOYODA	9	0.000	∗
			621	S-T	8	0.000	∗
			616	L-MM1	13	0.000	∗
			622	FOX	6	0.000	∗
			553	KOCHEN	41	0.000	∗
	1,1,1,1,2	NG V2	587	TOYODA	30	0.000	∗
			604	S-T	20	0.000	∗
			593	L-MM1	30	0.000	∗
			623	FOX	5	0.000	∗
			441	KOCHEN	90	0.000	∗
	1,1,1,2,2	NG V2	568	TOYODA	31	0.000	∗
			575	S-T	27	0.000	∗
			554	L-MM1	61	0.000	∗
			622	FOX	7	0.000	∗
			366	KOCHEN	116	0.000	∗
	1,1,2,2,2	NG V2	552	TOYODA	53	0.000	∗
			480	S-T	78	0.000	∗
			511	L-MM1	91	0.000	∗
			620	FOX	4	0.000	∗
			308	KOCHEN	136	0.000	∗
	1,2,2,2,2	NG V2	544	TOYODA	64	0.000	∗
			302	S-T	223	0.000	∗
			426	L-MM1	156	0.000	∗
			598	FOX	2	0.000	∗
			249	KOCHEN	193	0.000	∗
	2,2,2,2,2	NG V2	597	TOYODA	10	0.000	∗
			602	S-T	8	0.000	∗
			582	L-MM1	26	0.000	∗
			582	FOX	34	0.000	∗
			472	KOCHEN	53	0.000	∗

U ~ the number times best heuristic is better than compared heuristic
∗ indicates NG V2 as the best

Table E.7 Number of Times Best by NG V2 and Each Heuristic under 2KP Correlation Structures

	NG V2	TOYODA	S-T	L-M M1	FOX	KOCHEN	Sign Test
2,2,2	**4**	0	0	0	2	1	∗
2,1,1	**7**	1	0	1	4	0	∗
2,0,0	**8**	0	0	0	3	0	∗
2,-1,-1	**10**	0	0	0	1	2	∗
2,-2,-2	**16**	0	0	0	0	2	∗
1,2,1	**5**	1	0	0	7	0	
1,1,2	**0**	0	0	0	6	0	
1,1,1	**3**	0	4	0	0	2	
1,1,0	**3**	0	4	0	0	1	
1,0,1	**6**	0	0	0	3	1	∗
1,0,0	**14**	0	1	0	0	6	∗
1,0,-1	**5**	0	5	0	0	3	∗
1,-1,0	**4**	1	1	0	3	2	
1,-1,-1	**14**	0	4	0	0	3	∗
1,-1,-2	**6**	0	4	0	0	2	
1,-2,-1	**6**	0	1	0	0	1	∗
0,2,0	**9**	0	1	0	1	0	∗
0,1,1	**6**	0	1	0	1	0	∗
0,1,0	**14**	0	1	0	0	5	
0,1,-1	**3**	0	6	0	0	1	∗
0,0,2	**0**	2	0	1	1	0	
0,0,1	**4**	2	1	2	1	1	
0,0,0	**11**	1	3	0	0	2	∗
0,0,-1	**11**	1	2	0	0	3	∗
0,0,-2	**1**	0	7	0	0	2	
0,-1,1	**2**	3	0	0	1	1	
0,-1,0	**9**	0	2	2	1	3	∗
0,-1,-1	**5**	0	1	2	0	0	∗
0,-2,0	**5**	1	0	0	0	0	∗
-1,2,-1	**9**	1	1	1	1	0	∗
-1,1,0	**5**	0	0	2	1	1	∗
-1,1,-1	**12**	0	3	1	1	6	∗
-1,1,-2	**8**	1	4	1	0	1	∗
-1,0,1	**2**	1	0	0	3	0	
-1,0,0	**7**	1	1	2	0	3	
-1,0,-1	**3**	0	2	1	0	1	
-1,-1,2	**0**	1	0	3	1	0	
-1,-1,1	**3**	2	1	2	1	2	
-1,-1,0	**2**	0	1	2	0	1	
-1,-2,1	**2**	0	1	0	2	1	
-2,2,-2	**11**	0	0	0	2	0	∗
-2,1,-1	**3**	0	0	1	2	4	
-2,0,0	**5**	0	1	1	0	1	∗
-2,-1,1	**4**	0	0	0	3	0	
-2,-2,2	**0**	0	0	0	0	0	

(∗ indicates NG V2 as the best, Reject Region: $\alpha = 0.1$)

Table E.8 Number of Times Best by NG V2 and Each Heuristic under 5KP Correlation Structures

Correlation	**NG V2**	TOYODA	S-T	L-M M1	FOX	KOCHEN	Sign Test
-2,-2,-2,-2,-2	**6**	0	0	2	4	0	*
-2,-2,-2,-2,-1	**5**	1	1	2	5	1	
-2,-2,-2,-2,0	**9**	1	1	5	4	2	*
-2,-2,-2,-2,1	**9**	0	3	3	2	1	
-2,-2,-2,-2,2	**10**	1	0	5	3	2	*
-2,-2,-2,-1,-1	**6**	0	0	3	2	0	*
-2,-2,-2,-1,0	**8**	1	0	6	1	0	*
-2,-2,-2,-1,1	**14**	0	0	7	1	1	*
-2,-2,-2,-1,2	**12**	1	0	3	4	2	*
-2,-2,-2,0,0	**11**	1	0	5	0	4	*
-2,-2,-2,0,1	**14**	0	3	2	1	1	*
-2,-2,-2,0,2	**17**	0	0	3	0	3	*
-2,-2,-2,1,1	**16**	0	4	2	0	5	*
-2,-2,-2,1,2	**19**	0	1	3	0	3	*
-2,-2,-2,2,2	**20**	0	1	2	0	4	*
-2,-2,-1,-1,-1	**6**	0	1	4	1	4	*
-2,-2,-1,-1,0	**12**	2	0	5	1	1	*
-2,-2,-1,-1,1	**16**	0	1	3	0	4	*
-2,-2,-1,-1,2	**19**	0	0	3	0	3	*
-2,-2,-1,0,0	**15**	0	1	2	1	0	*
-2,-2,-1,0,1	**12**	0	2	7	0	2	*
-2,-2,-1,0,2	**13**	0	0	3	2	4	*
-2,-2,-1,1,1	**17**	0	1	6	0	3	*
-2,-2,-1,1,2	**20**	0	1	2	1	2	*
-2,-2,-1,2,2	**20**	0	0	1	0	4	*
-2,-2,0,0,0	**11**	0	1	5	1	2	*
-2,-2,0,0,1	**21**	1	2	0	0	1	*
-2,-2,0,0,2	**21**	0	1	0	0	5	*
-2,-2,0,1,1	**15**	1	1	2	0	6	*
-2,-2,0,1,2	**24**	0	1	1	0	2	*
-2,-2,0,2,2	**20**	1	0	2	0	5	*
-2,-2,1,1,1	**15**	0	2	4	0	5	*
-2,-2,1,1,2	**20**	0	0	1	0	4	*
-2,-2,1,2,2	**20**	0	1	0	1	3	*
-2,-2,2,2,2	**22**	1	1	1	0	4	*
-2,-1,-1,-1,-1	**9**	0	0	5	0	2	*
-2,-1,-1,-1,0	**14**	0	1	5	1	1	*
-2,-1,-1,-1,1	**16**	0	2	3	1	1	*
-2,-1,-1,-1,2	**16**	0	1	1	1	5	*
-2,-1,-1,0,0	**17**	0	0	3	0	3	*
-2,-1,-1,0,1	**15**	0	2	2	0	5	*
-2,-1,-1,0,2	**16**	2	1	3	1	2	*
-2,-1,-1,1,1	**19**	0	3	1	0	1	*
-2,-1,-1,1,2	**19**	0	1	1	0	4	*
-2,-1,-1,2,2	**18**	0	2	2	0	2	*
-2,-1,0,0,0	**11**	0	1	1	0	6	*
-2,-1,0,0,1	**18**	0	0	3	0	1	*
-2,-1,0,0,2	**17**	0	0	3	1	4	*
-2,-1,0,1,1	**16**	0	0	4	0	3	*
-2,-1,0,1,2	**19**	0	0	1	0	5	*
-2,-1,0,2,2	**19**	2	0	3	0	6	*
-2,-1,1,1,1	**20**	0	2	1	0	2	*
-2,-1,1,1,2	**21**	0	1	2	0	4	*
-2,-1,1,2,2	**26**	0	0	0	0	2	*
-2,-1,2,2,2	**19**	0	2	3	0	4	*
-2,0,0,0,0	**10**	0	2	0	0	4	*
-2,0,0,0,1	**16**	0	1	3	0	4	*
-2,0,0,0,2	**19**	0	1	0	0	3	*
-2,0,0,1,1	**18**	0	1	1	0	4	*
-2,0,0,1,2	**19**	0	0	2	0	5	*
-2,0,0,2,2	**21**	1	0	1	0	2	*
-2,0,1,1,1	**15**	0	1	0	0	7	*
-2,0,1,1,2	**22**	0	1	1	1	5	*

Correlation	NG V2	TOYODA	S-T	L-M M1	FOX	KOCHEN	Sign Test
-2,0,1,2,2	**20**	0	1	3	0	5	*
-2,0,2,2,2	**22**	0	0	1	0	5	*
-2,1,1,1,1	**14**	3	2	1	0	4	*
-2,1,1,1,2	**19**	0	0	1	0	4	*
-2,1,1,2,2	**23**	0	0	0	0	5	*
-2,1,2,2,2	**24**	0	1	2	0	1	*
-2,2,2,2,2	**17**	0	0	0	0	7	*
-1,-1,-1,-1,-1	**5**	1	1	1	0	4	
-1,-1,-1,-1,0	**6**	3	1	5	0	1	*
-1,-1,-1,-1,1	**16**	1	1	2	2	0	*
-1,-1,-1,-1,2	**11**	0	3	3	1	3	*
-1,-1,-1,0,0	**12**	1	1	3	0	4	*
-1,-1,-1,0,1	**12**	2	2	2	0	3	*
-1,-1,-1,0,2	**15**	0	4	4	1	2	*
-1,-1,-1,1,1	**22**	1	2	1	0	2	*
-1,-1,-1,1,2	**17**	2	1	1	0	2	*
-1,-1,-1,2,2	**17**	0	1	3	0	1	*
-1,-1,0,0,0	**14**	0	0	1	0	2	*
-1,-1,0,0,1	**15**	1	0	2	0	2	*
-1,-1,0,0,2	**12**	0	5	1	2	6	*
-1,-1,0,1,1	**17**	0	1	1	0	4	*
-1,-1,0,1,2	**17**	0	3	1	1	1	*
-1,-1,0,2,2	**18**	1	3	2	0	1	*
-1,-1,1,1,1	**17**	1	1	1	0	5	*
-1,-1,1,1,2	**22**	2	2	1	0	1	*
-1,-1,1,2,2	**21**	0	2	3	0	2	*
-1,-1,2,2,2	**20**	2	4	0	0	1	*
-1,0,0,0,0	**15**	0	1	0	0	3	*
-1,0,0,0,1	**16**	1	1	2	0	4	*
-1,0,0,0,2	**17**	0	0	0	0	7	*
-1,0,0,1,1	**19**	0	0	1	0	2	*
-1,0,0,1,2	**22**	0	2	1	0	2	*
-1,0,0,2,2	**18**	0	2	1	0	3	*
-1,0,1,1,1	**17**	0	2	0	0	6	*
-1,0,1,1,2	**24**	0	3	0	0	0	*
-1,0,1,2,2	**19**	0	2	0	0	4	*
-1,0,2,2,2	**20**	0	3	1	0	5	*
-1,1,1,1,1	**17**	1	3	0	0	6	*
-1,1,1,1,2	**23**	0	3	0	0	1	*
-1,1,1,2,2	**22**	0	3	1	0	3	*
-1,1,2,2,2	**22**	0	3	0	0	3	*
-1,2,2,2,2	**21**	1	4	1	0	1	*
0,0,0,0,0	**8**	2	2	1	0	4	
0,0,0,0,1	**14**	1	3	0	0	2	*
0,0,0,0,2	**15**	0	3	0	0	4	*
0,0,0,1,1	**20**	2	4	0	0	3	*
0,0,0,1,2	**22**	0	1	0	0	3	*
0,0,0,2,2	**19**	1	2	0	0	3	*
0,0,1,1,1	**16**	0	3	0	0	4	*
0,0,1,1,2	**19**	0	3	0	0	2	*
0,0,1,2,2	**19**	1	5	0	0	3	*
0,0,2,2,2	**22**	0	2	0	0	2	*
0,1,1,1,1	**19**	0	2	0	0	5	*
0,1,1,1,2	**22**	0	2	0	0	3	*
0,1,1,2,2	**23**	0	2	0	0	1	*
0,1,2,2,2	**21**	0	3	0	0	3	*
0,2,2,2,2	**23**	0	3	0	0	1	*
1,1,1,1,1	**12**	1	1	0	0	3	*
1,1,1,1,2	**16**	1	3	1	0	3	*
1,1,1,2,2	**19**	1	0	0	0	5	*
1,1,2,2,2	**23**	0	2	0	0	1	*
1,2,2,2,2	**25**	0	1	0	0	2	*
2,2,2,2,2	**11**	0	1	0	0	5	*

(* indicates NG V2 as the best, Reject Region: $\alpha = 0.1$)

Table E.9 Sign Test for NG V3 based on Constraint Slackness

Prob Type	Constraint Slackness	NG V3 Heuristic		vs. Legacy Heuristic		Pr(X ≥ U)	Reject Region (α = 0.1)
2KP	1, 1	NG V3	223	TOYODA	18	0.000	∗
			230	S-T	17	0.000	∗
			263	L-MM1	8	0.000	∗
			242	FOX	15	0.000	∗
			139	KOCHEN	49	0.000	∗
	1, 2	NG V3	222	TOYODA	24	0.000	∗
			103	S-T	69	0.006	∗
			249	L-MM1	11	0.000	∗
			174	FOX	7	0.000	∗
			116	KOCHEN	67	0.000	∗
	2, 1	NG V3	216	TOYODA	25	0.000	∗
			104	S-T	65	0.002	∗
			250	L-MM1	11	0.000	∗
			166	FOX	6	0.000	∗
			116	KOCHEN	59	0.000	∗
	2, 2	NG V3	208	TOYODA	16	0.000	∗
			213	S-T	14	0.000	∗
			241	L-MM1	15	0.000	∗
			187	FOX	28	0.000	∗
			125	KOCHEN	18	0.000	∗
5KP	1,1,1,1,1	NG V3	617	TOYODA	10	0.000	∗
			622	S-T	7	0.000	∗
			616	L-MM1	12	0.000	∗
			625	FOX	5	0.000	∗
			552	KOCHEN	49	0.000	∗
	1,1,1,1,2	NG V3	592	TOYODA	25	0.000	∗
			604	S-T	18	0.000	∗
			602	L-MM1	20	0.000	∗
			624	FOX	4	0.000	∗
			460	KOCHEN	89	0.000	∗
	1,1,1,2,2	NG V3	571	TOYODA	30	0.000	∗
			577	S-T	24	0.000	∗
			562	L-MM1	48	0.000	∗
			622	FOX	5	0.000	∗
			396	KOCHEN	120	0.000	∗
	1,1,2,2,2	NG V3	558	TOYODA	52	0.000	∗
			491	S-T	69	0.000	∗
			526	L-MM1	74	0.000	∗
			620	FOX	4	0.000	∗
			350	KOCHEN	135	0.000	∗
	1,2,2,2,2	NG V3	562	TOYODA	46	0.000	∗
			348	S-T	166	0.000	∗
			460	L-MM1	110	0.000	∗
			596	FOX	1	0.000	∗
			330	KOCHEN	164	0.000	∗
	2,2,2,2,2	NG V3	597	TOYODA	10	0.000	∗
			601	S-T	9	0.000	∗
			586	L-MM1	21	0.000	∗
			594	FOX	19	0.000	∗
			484	KOCHEN	58	0.000	∗

U ~ the number times best heuristic is better than compared heuristic
∗ indicates NG V3 as the best

Table E.10 Number of Times Best by NG V3 and Each heuristic under 2KP Correlation Structures

	NG V3	TOYODA	S-T	L-M M1	FOX	KOCHEN	Sign Test
2,2,2	**4**	0	0	0	2	1	∗
2,1,1	**8**	1	0	0	4	0	∗
2,0,0	**9**	0	0	0	2	0	∗
2,-1,-1	**9**	0	0	0	1	2	∗
2,-2,-2	**13**	0	1	0	0	2	
1,2,1	**6**	1	0	0	6	0	∗
1,1,2	**0**	0	0	0	6	0	
1,1,1	**3**	0	4	0	0	2	
1,1,0	**4**	0	4	0	0	0	
1,0,1	**5**	0	0	0	3	2	∗
1,0,0	**15**	0	1	0	0	5	∗
1,0,-1	**4**	0	2	0	0	5	
1,-1,0	**4**	1	1	0	2	5	
1,-1,-1	**19**	0	3	0	0	3	∗
1,-1,-2	**8**	0	2	0	0	3	∗
1,-2,-1	**7**	0	1	0	0	1	∗
0,2,0	**11**	0	0	0	0	0	∗
0,1,1	**6**	0	1	0	1	0	∗
0,1,0	**18**	0	1	1	0	5	∗
0,1,-1	**3**	0	3	0	0	2	∗
0,0,2	**0**	2	0	1	1	0	
0,0,1	**3**	1	1	2	1	2	
0,0,0	**12**	0	4	0	0	3	∗
0,0,-1	**14**	1	2	0	0	3	∗
0,0,-2	**5**	0	2	0	0	2	
0,-1,1	**3**	3	0	0	1	1	
0,-1,0	**10**	0	2	3	1	4	∗
0,-1,-1	**7**	1	1	2	0	0	
0,-2,0	**4**	1	0	0	0	2	∗
-1,2,-1	**9**	1	1	1	1	0	∗
-1,1,0	**6**	0	0	2	0	2	
-1,1,-1	**14**	0	2	0	1	7	∗
-1,1,-2	**8**	1	2	1	0	4	
-1,0,1	**2**	1	0	0	2	0	
-1,0,0	**9**	0	2	1	0	4	
-1,0,-1	**4**	1	2	1	0	2	
-1,-1,2	**0**	1	0	3	1	0	
-1,-1,1	**4**	2	1	2	1	2	
-1,-1,0	**3**	0	1	2	0	1	
-1,-2,1	**2**	0	1	0	2	1	
-2,2,-2	**10**	0	0	1	1	0	
-2,1,-1	**4**	0	0	1	1	4	
-2,0,0	**4**	0	1	0	0	1	∗
-2,-1,1	**5**	0	0	0	1	0	∗
-2,-2,2	**0**	0	0	0	0	0	

(∗ indicates NG V3 as the best, Reject Region: $\alpha = 0.1$)

Table E.11 Number of Times Best by NG V3 and Each Heuristic under 5KP Correlation Structures

Correlation	**NG V3**	TOYODA	S-T	L-M M1	FOX	KOCHEN	Sign Test
-2,-2,-2,-2,-2	**6**	0	0	2	4	0	*
-2,-2,-2,-2,-1	**5**	1	1	2	5	2	
-2,-2,-2,-2,0	**8**	1	1	5	4	2	*
-2,-2,-2,-2,1	**11**	0	2	3	1	1	*
-2,-2,-2,-2,2	**13**	0	0	4	1	1	*
-2,-2,-2,-1,-1	**7**	0	0	3	1	0	*
-2,-2,-2,-1,0	**12**	1	0	4	1	0	*
-2,-2,-2,-1,1	**14**	0	0	7	0	2	*
-2,-2,-2,-1,2	**13**	1	0	2	2	1	*
-2,-2,-2,0,0	**12**	1	0	5	0	4	*
-2,-2,-2,0,1	**16**	0	1	1	1	1	*
-2,-2,-2,0,2	**19**	0	0	2	0	3	*
-2,-2,-2,1,1	**16**	0	3	2	0	5	*
-2,-2,-2,1,2	**19**	0	1	3	0	4	*
-2,-2,-2,2,2	**20**	0	1	2	0	4	*
-2,-2,-1,-1,-1	**7**	0	1	4	1	5	*
-2,-2,-1,-1,0	**14**	2	0	2	1	3	*
-2,-2,-1,-1,1	**16**	0	1	3	0	6	*
-2,-2,-1,-1,2	**19**	0	0	3	0	5	*
-2,-2,-1,0,0	**16**	0	1	2	0	2	*
-2,-2,-1,0,1	**16**	0	1	4	0	3	*
-2,-2,-1,0,2	**21**	0	0	2	0	4	*
-2,-2,-1,1,1	**17**	0	1	5	0	4	*
-2,-2,-1,1,2	**22**	0	1	1	0	3	*
-2,-2,-1,2,2	**22**	0	0	0	0	2	*
-2,-2,0,0,0	**11**	0	1	4	1	4	*
-2,-2,0,0,1	**22**	1	2	0	0	0	*
-2,-2,0,0,2	**21**	0	1	0	0	6	*
-2,-2,0,1,1	**17**	1	1	1	0	7	*
-2,-2,0,1,2	**23**	0	1	1	0	2	*
-2,-2,0,2,2	**21**	1	0	2	0	4	*
-2,-2,1,1,1	**16**	0	2	3	0	5	*
-2,-2,1,1,2	**21**	0	0	1	0	4	*
-2,-2,1,2,2	**21**	0	1	0	0	3	*
-2,-2,2,2,2	**20**	1	1	1	0	6	*
-2,-1,-1,-1,-1	**9**	0	0	6	0	1	*
-2,-1,-1,-1,0	**11**	0	1	7	1	1	*
-2,-1,-1,-1,1	**18**	0	2	3	1	1	*
-2,-1,-1,-1,2	**18**	0	1	1	0	5	*
-2,-1,-1,0,0	**19**	0	0	2	0	4	*
-2,-1,-1,0,1	**16**	0	2	1	0	6	*
-2,-1,-1,0,2	**15**	2	1	3	1	5	*
-2,-1,-1,1,1	**21**	0	2	0	0	2	*
-2,-1,-1,1,2	**21**	0	0	1	0	6	*
-2,-1,-1,2,2	**20**	0	2	1	0	1	*
-2,-1,0,0,0	**14**	0	0	1	0	6	*
-2,-1,0,0,1	**18**	0	0	3	0	1	*
-2,-1,0,0,2	**20**	0	0	2	0	6	*
-2,-1,0,1,1	**18**	0	0	2	0	3	*
-2,-1,0,1,2	**19**	0	0	0	0	6	*
-2,-1,0,2,2	**19**	2	0	2	0	5	*
-2,-1,1,1,1	**19**	0	1	1	0	2	*
-2,-1,1,1,2	**21**	0	1	0	0	4	*
-2,-1,1,2,2	**25**	0	0	0	0	3	*
-2,-1,2,2,2	**20**	0	1	1	0	7	*
-2,0,0,0,0	**9**	0	2	0	0	6	
-2,0,0,0,1	**18**	0	1	0	0	4	*
-2,0,0,0,2	**19**	0	1	0	0	4	*
-2,0,0,1,1	**19**	0	1	1	0	4	*
-2,0,0,1,2	**21**	0	0	2	0	5	*
-2,0,0,2,2	**22**	1	0	0	0	3	*
-2,0,1,1,1	**17**	0	1	0	0	8	*

Correlation	**NG V3**	TOYODA	S-T	L-M M1	FOX	KOCHEN	Sign Test
-2,0,1,1,2	**24**	0	1	1	0	4	*
-2,0,1,2,2	**23**	0	1	2	0	2	*
-2,0,2,2,2	**22**	0	0	0	0	5	*
-2,1,1,1,1	**9**	3	2	0	0	10	
-2,1,1,1,2	**20**	0	0	1	0	6	*
-2,1,1,2,2	**22**	0	0	0	0	6	*
-2,1,2,2,2	**24**	0	1	2	0	3	*
-2,2,2,2,2	**13**	0	0	0	0	11	
-1,-1,-1,-1,-1	**6**	1	1	1	0	4	
-1,-1,-1,-1,0	**6**	3	1	5	0	0	*
-1,-1,-1,-1,1	**17**	1	0	2	2	1	*
-1,-1,-1,-1,2	**15**	0	2	2	0	5	*
-1,-1,-1,0,0	**15**	1	1	3	0	4	*
-1,-1,-1,0,1	**17**	2	2	1	0	2	*
-1,-1,-1,0,2	**18**	0	4	1	0	1	*
-1,-1,-1,1,1	**21**	1	2	0	0	3	*
-1,-1,-1,1,2	**19**	2	1	1	0	2	*
-1,-1,-1,2,2	**19**	0	1	3	0	1	*
-1,-1,0,0,0	**16**	0	0	1	0	1	*
-1,-1,0,0,1	**18**	1	0	1	0	3	*
-1,-1,0,0,2	**17**	0	2	1	2	6	*
-1,-1,0,1,1	**24**	0	2	0	0	2	*
-1,-1,0,1,2	**19**	0	1	0	1	2	*
-1,-1,0,2,2	**20**	1	2	1	0	0	*
-1,-1,1,1,1	**18**	1	0	1	0	5	*
-1,-1,1,1,2	**24**	1	1	0	0	1	*
-1,-1,1,2,2	**21**	0	2	3	0	1	*
-1,-1,2,2,2	**21**	2	3	0	0	1	*
-1,0,0,0,0	**17**	0	1	0	0	5	*
-1,0,0,0,1	**19**	1	0	1	0	4	*
-1,0,0,0,2	**18**	0	1	0	0	6	*
-1,0,0,1,1	**21**	0	0	1	0	2	*
-1,0,0,1,2	**23**	0	2	1	0	2	*
-1,0,0,2,2	**20**	0	1	1	0	3	*
-1,0,1,1,1	**19**	0	1	0	0	5	*
-1,0,1,1,2	**26**	0	1	0	0	1	*
-1,0,1,2,2	**23**	0	1	0	0	4	*
-1,0,2,2,2	**22**	0	2	0	0	6	*
-1,1,1,1,1	**18**	1	1	0	0	5	*
-1,1,1,1,2	**23**	0	3	0	0	2	*
-1,1,1,2,2	**22**	0	3	1	0	3	*
-1,1,2,2,2	**24**	0	2	0	0	1	*
-1,2,2,2,2	**25**	1	3	0	0	1	*
0,0,0,0,0	**6**	2	2	1	0	6	
0,0,0,0,1	**18**	0	3	0	0	3	*
0,0,0,0,2	**15**	0	2	0	0	4	*
0,0,0,1,1	**22**	2	3	0	0	3	*
0,0,0,1,2	**23**	0	2	0	0	2	*
0,0,0,2,2	**19**	1	1	0	0	2	*
0,0,1,1,1	**13**	0	2	0	0	11	*
0,0,1,1,2	**19**	1	2	0	0	3	*
0,0,1,2,2	**20**	1	3	0	0	3	*
0,0,2,2,2	**22**	0	1	0	0	1	*
0,1,1,1,1	**24**	0	1	0	0	2	*
0,1,1,1,2	**23**	0	0	0	0	6	*
0,1,1,2,2	**26**	0	1	0	0	1	*
0,1,2,2,2	**25**	0	1	0	0	1	*
0,2,2,2,2	**23**	0	2	0	0	3	*
1,1,1,1,1	**12**	1	1	0	0	3	*
1,1,1,1,2	**17**	2	2	0	0	2	*
1,1,1,2,2	**20**	0	2	0	0	2	*
1,1,2,2,2	**23**	0	2	0	0	2	*
1,2,2,2,2	**25**	0	1	0	0	0	*
2,2,2,2,2	**11**	0	0	0	0	3	*

(∗ indicates NG V3 as the best, Reject Region: $\alpha = 0.1$)

Table E.12 Sign Test for New Reduction Heuristic (Core Solved by NG V3)

Prob Type	Constraint Slackness	NG V3 Heuristic		vs. Legacy Heuristic		Pr(X ≥ U)	Reject Region (α = 0.1)
2KP	1, 1	New Reduction	220	TOYODA	36	0.000	∗
			232	S-T	28	0.000	∗
			255	L-MM1	18	0.000	∗
			239	FOX	18	0.000	∗
			135	KOCHEN	86	0.001	∗
	1, 2	New Reduction	219	TOYODA	29	0.000	∗
			102	S-T	73	0.017	∗
			248	L-MM1	11	0.000	∗
			173	FOX	13	0.000	∗
			114	KOCHEN	73	0.002	∗
	2, 1	New Reduction	215	TOYODA	27	0.000	∗
			103	S-T	66	0.003	∗
			248	L-MM1	10	0.000	∗
			167	FOX	9	0.000	∗
			116	KOCHEN	61	0.000	∗
	2, 2	New Reduction	206	TOYODA	23	0.000	∗
			210	S-T	21	0.000	∗
			230	L-MM1	22	0.000	∗
			172	FOX	20	0.000	∗
			134	KOCHEN	51	0.000	∗
5KP	1,1,1,1,1	New Reduction	614	TOYODA	12	0.000	∗
			620	S-T	8	0.000	∗
			614	L-MM1	14	0.000	∗
			628	FOX	2	0.000	∗
			553	KOCHEN	64	0.000	∗
	1,1,1,1,2	New Reduction	588	TOYODA	40	0.000	∗
			604	S-T	26	0.000	∗
			586	L-MM1	31	0.000	∗
			622	FOX	3	0.000	∗
			460	KOCHEN	146	0.000	∗
	1,1,1,2,2	New Reduction	562	TOYODA	52	0.000	∗
			573	S-T	45	0.000	∗
			546	L-MM1	65	0.000	∗
			621	FOX	4	0.000	∗
			391	KOCHEN	193	0.000	∗
	1,1,2,2,2	New Reduction	540	TOYODA	76	0.000	∗
			483	S-T	99	0.000	∗
			490	L-MM1	100	0.000	∗
			619	FOX	6	0.000	∗
			328	KOCHEN	222	0.000	∗
	1,2,2,2,2	New Reduction	565	TOYODA	54	0.000	∗
			367	S-T	153	0.000	∗
			451	L-MM1	113	0.000	∗
			595	FOX	2	0.000	∗
			349	KOCHEN	195	0.000	∗
	2,2,2,2,2	New Reduction	594	TOYODA	17	0.000	∗
			601	S-T	14	0.000	∗
			586	L-MM1	24	0.000	∗
			591	FOX	9	0.000	∗
			495	KOCHEN	84	0.000	∗

U ~ the number times best heuristic is better than compared heuristic
∗ indicates New Reduction Heuristic as the best

Table E.13 Number of Times Best by New Reduction Heuristic (Core Solved by NG V3) and Each Heuristic under 2KP Correlation Structure

	New Reduction	TOYODA	S-T	L-M M1	FOX	KOCHEN	Sign Test
2,2,2	**5**	0	0	0	3	1	
2,1,1	**10**	1	0	1	3	0	∗
2,0,0	**8**	0	0	0	3	0	∗
2,-1,-1	**8**	0	0	0	2	2	∗
2,-2,-2	**13**	0	0	0	1	3	∗
1,2,1	**5**	1	0	0	4	0	∗
1,1,2	**1**	0	0	0	6	0	
1,1,1	**3**	0	4	0	0	1	
1,1,0	**5**	0	4	0	0	2	
1,0,1	**6**	0	0	0	1	3	∗
1,0,0	**16**	0	1	0	0	7	∗
1,0,-1	**6**	0	3	0	0	5	
1,-1,0	**3**	1	1	0	0	6	
1,-1,-1	**14**	0	2	0	0	6	∗
1,-1,-2	**8**	0	1	0	0	5	
1,-2,-1	**5**	0	1	0	1	1	∗
0,2,0	**10**	0	0	0	2	0	∗
0,1,1	**6**	0	1	0	0	0	∗
0,1,0	**15**	0	1	0	0	7	
0,1,-1	**6**	0	3	0	0	5	
0,0,2	**0**	2	0	1	0	0	
0,0,1	**3**	1	1	2	1	1	
0,0,0	**12**	0	4	0	0	5	∗
0,0,-1	**12**	1	2	0	0	6	
0,0,-2	**4**	0	2	1	0	2	
0,-1,1	**2**	3	0	0	1	3	
0,-1,0	**10**	0	2	3	0	6	
0,-1,-1	**5**	1	1	2	0	2	
0,-2,0	**2**	1	0	0	1	3	∗
-1,2,-1	**9**	1	1	1	2	0	∗
-1,1,0	**7**	0	0	2	0	1	∗
-1,1,-1	**15**	0	2	0	1	10	∗
-1,1,-2	**4**	1	2	2	0	6	
-1,0,1	**3**	1	0	0	1	0	∗
-1,0,0	**10**	0	2	0	0	5	
-1,0,-1	**4**	1	2	1	0	5	
-1,-1,2	**0**	1	0	3	1	0	
-1,-1,1	**4**	2	1	1	1	3	
-1,-1,0	**3**	1	1	2	0	2	
-1,-2,1	**2**	0	1	0	1	2	
-2,2,-2	**7**	0	0	1	3	2	∗
-2,1,-1	**4**	0	0	0	0	4	∗
-2,0,0	**2**	0	1	0	1	2	∗
-2,-1,1	**3**	0	0	0	0	1	∗
-2,-2,2	**0**	0	0	0	0	0	

(∗ indicates New Reduction Heuristic as the best, Reject Region: $\alpha = 0.1$)

Table E.14 Number of Times Best by New Reduction Heuristic (Core Solved by NG V3) and Each Heuristic under 5KP Correlation Structure

Correlation	NR	TOYODA	S-T	L-M M1	FOX	KOCHEN	Sign Test
-2,-2,-2,-2,-2	**8**	0	0	1	4	1	*
-2,-2,-2,-2,-1	**5**	1	1	1	2	2	
-2,-2,-2,-2,0	**10**	1	1	2	1	4	*
-2,-2,-2,-2,1	**12**	0	2	1	0	0	*
-2,-2,-2,-2,2	**13**	0	0	4	0	4	*
-2,-2,-2,-1,-1	**11**	0	0	3	0	4	*
-2,-2,-2,-1,0	**13**	0	0	4	0	2	*
-2,-2,-2,-1,1	**13**	0	0	4	0	4	*
-2,-2,-2,-1,2	**13**	0	0	3	2	4	*
-2,-2,-2,0,0	**13**	1	1	4	0	5	*
-2,-2,-2,0,1	**18**	0	2	1	0	2	*
-2,-2,-2,0,2	**22**	1	0	2	0	3	*
-2,-2,-2,1,1	**17**	0	2	1	0	6	*
-2,-2,-2,1,2	**22**	0	1	2	0	5	*
-2,-2,-2,2,2	**17**	1	1	2	0	6	*
-2,-2,-1,-1,-1	**9**	1	1	2	1	9	
-2,-2,-1,-1,0	**12**	2	0	3	1	4	*
-2,-2,-1,-1,1	**17**	0	1	1	0	8	*
-2,-2,-1,-1,2	**18**	0	0	2	0	5	*
-2,-2,-1,0,0	**14**	0	1	3	1	5	*
-2,-2,-1,0,1	**16**	0	1	6	0	3	*
-2,-2,-1,0,2	**16**	0	0	2	0	8	
-2,-2,-1,1,1	**19**	0	1	2	0	3	*
-2,-2,-1,1,2	**19**	1	1	0	0	5	*
-2,-2,-1,2,2	**18**	0	0	1	0	8	*
-2,-2,0,0,0	**9**	0	1	4	1	8	
-2,-2,0,0,1	**18**	1	1	0	0	4	*
-2,-2,0,0,2	**20**	0	1	0	0	9	*
-2,-2,0,1,1	**15**	1	1	1	0	7	*
-2,-2,0,1,2	**22**	0	0	3	0	3	*
-2,-2,0,2,2	**22**	1	0	1	0	3	*
-2,-2,1,1,1	**13**	0	2	2	0	7	*
-2,-2,1,1,2	**21**	0	0	0	0	7	*
-2,-2,1,2,2	**22**	0	1	0	0	5	*
-2,-2,2,2,2	**18**	1	1	2	0	7	*
-2,-1,-1,-1,-1	**11**	0	0	7	0	2	*
-2,-1,-1,-1,0	**15**	0	1	7	0	3	*
-2,-1,-1,-1,1	**15**	0	1	4	1	3	*
-2,-1,-1,-1,2	**16**	1	1	1	0	7	*
-2,-1,-1,0,0	**15**	0	0	3	0	5	*
-2,-1,-1,0,1	**16**	0	2	3	0	5	*
-2,-1,-1,0,2	**15**	1	1	3	2	4	*
-2,-1,-1,1,1	**19**	0	2	1	0	2	*
-2,-1,-1,1,2	**19**	0	0	1	0	7	*
-2,-1,-1,2,2	**22**	0	1	0	0	2	*
-2,-1,0,0,0	**15**	0	0	3	0	7	*
-2,-1,0,0,1	**20**	0	0	2	0	5	*
-2,-1,0,0,2	**22**	0	0	1	0	4	*
-2,-1,0,1,1	**17**	0	0	3	0	5	*
-2,-1,0,1,2	**21**	0	0	2	0	6	*
-2,-1,0,2,2	**21**	1	0	2	0	5	*
-2,-1,1,1,1	**19**	0	0	1	0	3	*
-2,-1,1,1,2	**22**	0	1	2	0	4	*
-2,-1,1,2,2	**25**	0	0	0	0	4	*
-2,-1,2,2,2	**22**	0	0	1	0	5	*
-2,0,0,0,0	**9**	0	1	0	1	7	
-2,0,0,0,1	**18**	0	0	2	0	7	*
-2,0,0,0,2	**17**	0	0	0	0	7	*
-2,0,0,1,1	**16**	0	1	1	0	9	*
-2,0,0,1,2	**17**	1	1	3	0	6	*
-2,0,0,2,2	**22**	1	0	1	0	4	*
-2,0,1,1,1	**16**	0	1	0	0	9	

Correlation	**NR**	TOYODA	S-T	L-M M1	FOX	KOCHEN	Sign Test
-2,0,1,1,2	**22**	0	1	1	0	6	*
-2,0,1,2,2	**24**	0	1	2	0	3	*
-2,0,2,2,2	**24**	0	0	0	0	3	*
-2,1,1,1,1	**7**	3	1	1	0	13	
-2,1,1,1,2	**22**	0	0	1	0	7	*
-2,1,1,2,2	**23**	0	0	0	0	5	*
-2,1,2,2,2	**20**	0	1	1	0	8	*
-2,2,2,2,2	**16**	0	0	0	0	11	
-1,-1,-1,-1,-1	**7**	1	1	1	0	6	
-1,-1,-1,-1,0	**10**	2	1	3	0	2	*
-1,-1,-1,-1,1	**17**	0	1	1	0	5	*
-1,-1,-1,-1,2	**18**	0	2	1	0	4	*
-1,-1,-1,0,0	**15**	1	1	1	0	8	*
-1,-1,-1,0,1	**14**	2	2	3	0	3	*
-1,-1,-1,0,2	**19**	0	3	2	0	4	*
-1,-1,-1,1,1	**19**	1	2	1	0	4	*
-1,-1,-1,1,2	**20**	2	1	1	0	3	*
-1,-1,-1,2,2	**19**	1	1	3	0	2	*
-1,-1,0,0,0	**16**	0	0	2	0	3	*
-1,-1,0,0,1	**16**	1	0	1	0	4	*
-1,-1,0,0,2	**17**	0	2	1	1	4	*
-1,-1,0,1,1	**21**	0	3	1	0	4	*
-1,-1,0,1,2	**19**	0	1	1	1	4	*
-1,-1,0,2,2	**21**	1	2	0	0	3	*
-1,-1,1,1,1	**23**	1	0	0	0	4	*
-1,-1,1,1,2	**24**	1	1	1	0	3	*
-1,-1,1,2,2	**20**	0	1	3	0	3	*
-1,-1,2,2,2	**21**	1	1	0	0	2	*
-1,0,0,0,0	**15**	0	1	0	0	9	*
-1,0,0,0,1	**22**	0	0	0	0	4	*
-1,0,0,0,2	**17**	0	1	0	0	10	
-1,0,0,1,1	**19**	0	0	1	0	4	*
-1,0,0,1,2	**20**	0	2	1	0	5	*
-1,0,0,2,2	**21**	0	1	1	0	4	*
-1,0,1,1,1	**18**	0	1	0	0	10	*
-1,0,1,1,2	**26**	0	0	0	0	3	*
-1,0,1,2,2	**23**	0	0	0	0	3	*
-1,0,2,2,2	**22**	0	2	0	0	5	*
-1,1,1,1,1	**17**	1	1	0	0	8	*
-1,1,1,1,2	**23**	0	1	0	0	2	*
-1,1,1,2,2	**21**	0	2	1	0	6	*
-1,1,2,2,2	**24**	0	1	0	0	3	*
-1,2,2,2,2	**24**	0	1	0	0	4	*
0,0,0,0,0	**8**	1	2	1	0	9	
0,0,0,0,1	**21**	1	3	0	0	3	*
0,0,0,0,2	**16**	0	1	0	0	6	*
0,0,0,1,1	**22**	2	4	0	0	2	*
0,0,0,1,2	**22**	0	2	0	0	4	*
0,0,0,2,2	**18**	0	1	0	0	5	*
0,0,1,1,1	**14**	0	2	1	0	9	*
0,0,1,1,2	**18**	1	2	0	0	4	*
0,0,1,2,2	**20**	1	2	0	0	3	*
0,0,2,2,2	**23**	0	1	0	0	1	*
0,1,1,1,1	**19**	0	1	0	0	7	*
0,1,1,1,2	**23**	0	0	0	0	6	*
0,1,1,2,2	**25**	0	1	0	0	2	*
0,1,2,2,2	**22**	0	1	0	0	4	*
0,2,2,2,2	**23**	0	2	0	0	3	*
1,1,1,1,1	**12**	1	1	0	0	10	
1,1,1,1,2	**18**	1	2	0	0	3	*
1,1,1,2,2	**20**	1	1	0	0	5	*
1,1,2,2,2	**24**	0	2	0	0	1	*
1,2,2,2,2	**22**	0	1	0	0	3	*
2,2,2,2,2	**11**	0	1	0	0	10	

(∗ indicates New Reduction Heuristic as the best, Reject Region: $\alpha = 0.1$)

Bibliography

Aboudi, R., and Jörnsten, K. "Tabu Search for General Zero-One Integer Programs Using the Pivot and Complement Heuristic," *ORSA Journal on Computing,* 6 (1): 82-93 (1994).

Akcay, Y., Li, H., and Xu, S. "An Approximate Algorithm for the General Multidimensional Knapsack Problem" *Washington State University Mathematical Technical Report, 2002-002,* 2002.

Averbakh, I. "Probabilistic Properties of the Dual Structure of the Multidimensional Knapsack Problem and Fast Statistically Efficient Algorithms," *Mathematical Programming,* 65: 311-330 (1994).

Balas, E., and Martin, C. "Pivot and Complement – A Heuristic for 0 – 1 Programming," *Management Science,* 26 (1): 86-96 (1980).

Balas, E., and Zemel, E. "An Algorithm for Large Zero – One Knapsack Problem," *Operations Research*, 28(5): 1130-1154 (1980).

Barr. R., Golden, B., Kelly, J., Resende, M., and Stewart., W. "Designing and Reporting on Computational Experiments with Heuristic Methods," *Journal of Heuristic*, 1(1): 9-32 (1995).

Beasley, J., Problems available at the website: http://mscmga.ms.ic.ac.uk/info.html, (2004).

Bertsimas, D. and Demir, R. "An Approximate Dynamic Programming Approach to Multidimensional Knapsack Problems," *Management Science*, 48(4): 550-565 (2002).

Cario, M., Clifford, J., Hill, R., Yang, J., Yang, K. and Reilly, C. "An Investigation of the Relationship between Problem Characteristics and Algorithm Performance: A Case Study of the Generalized Assignment Problems". *IIE Transactions on Operations Engineering*, 34(3):297-312 (2002).

Cherbaka, N. S., R. D. Meller, and K. P. Ellis. "Multidimensional Knapsack Problems and Their Application to Solving Manufacturing Insourcing Problems." *Proceedings of the Annual Industrial Engineering Research Conference*, Houston, TX, May 16-19, (2004).

Cho, Y. "Empirical Analysis of Various Multi-Dimensional Knapsack Heuristics," Master Thesis, Department of Operational Sciences, Air Force Institute of Technology, Dayton (2002).

Cho, Y., Moore, J. and Hill, R. “Developing A New Greedy Heuristic Based on Knowledge Gained via Structured Empirical Testing,” *International Journal of Industrial Engineering*, 10(4): 504-510 (2003b).

Cho, Y., Moore, J., Hill, R., and Reilly, C. “Exploiting Empirical Knowledge for Bi-Dimensional Knapsack Problem Heuristics,” *submitted to Computers & Industrial Engineering,* (2004)

Chu, P. and Beasley, J. “A Genetic Algorithm for the Multidimensional Knapsack Problem,” *Journal of Heuristics*, 4(1): 63-86 (1998).

Coy, S., Golden, B., Runger, G., and Wasil, E. “Using Experimental Design to Find Effective Parameter Settings for Heuristics,” *Journal of Heuristics*, 7: 77 - 97 (2001).

DeVries, S. and Vohra, R. “Combinatorial Auctions: A Survey,” *Northernwestern University Technical Report*, Evanston, IL (2000).

Drexl, A. “A Simulated Annealing Approach to the Multiconstraint Zero-One Knapsack Problem,” *Computing*, 40(1): 1-8 (1988).

Everett, H. “Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources,” *Operations Research*, 11: 399-417 (1963).

Ferreira, C., Grotschel, M., Kiefl, S., Krispenz, C., Martin, A., and Weismantel., R. “Some Integer Programs Arising in the Design of Mainframe Computers,” *ZOR-Methods Models Operations Research*, 38(1): 77-110 (1993).

Fisher, M., “The Lagrangian Relaxation Method for Solving Integer Programming Problems,” *Management Science* 27 (1): 1-18 (1981).

Fisher, M., “An Applications Oriented Guide to Lagrangian Relaxation,” *INTERFACES* 15 (2): 10-21, (1985).

Fox G. and Nachtsheim, C. “An Analysis of Six Greedy Selection Rules on a Class of Zero-One Integer Programming Models,” *Naval Research Logistics*, 37(2): 299-307 (1990).

Fox G. and Scudder, G. “A Heuristic with Tie Breaking for Certain 0 – 1 Integer Programming Models,” *Naval Research Logistics*, 32(4): 613-623 (1985).

Fréville, A, and Plateau, G. “Heuristics and Reduction Methods for Multiple Constraints 0 – 1 Linear Programming Problems,” *European Journal of Operational Research*, 24: 206-215 (1986).

Fréville, A, and Plateau, G. "Hard 0 – 1 Multiknapsack Test Problems for Size Reduction Methods," *Investigacion Operativa*, 1: 251-270 (1990).

Fréville, A, and Plateau, G. "An Exact Surrogate Dual Search for the 0 – 1 bidimensional knapsack problem," *European Journal of Operational Research*, 68: 413-421 (1993).

Fréville, A, and Plateau, G. "An Efficient Preprocessing Procedure for the Multidimensional 0 – 1 Knapsack Problem," *Discrete Applied Mathematics*, 49: 189-212 (1994).

Fréville, A, and Plateau, G. "The 0 – 1 Bidimensional Knapsack Problem: Towards an Efficient High Level Primitive Tool," *Journal of Heuristics*, 2: 147-167 (1996).

Frieze, A., and Clarke, M. "Approximation Algorithms for the M-Dimensional 0 – 1 Knapsack Problem: Worst-Case and Probabilistic Analyses," *European Journal of Operational Research*, 15: 100-109 (1984).

Garey M. and Johnson D. *Computers and intractability: a guide to the theory of NP-completeness*. San Francisco: W. H. Freeman, 1979.

Gavish, B. and Pirkul, H. "Efficient Algorithms for Solving Multiconstraint Zero-One Knapsack Problems to Optimality," *Mathematical Programming*, 31:78-105 (1985).

Geoffrion, A. "Integer Programming by Implicit Enumeration and Balas' Method," *SIAM Review*, 9:178-179 (1967).

Gilmore, P. and Gomery, R. "The Theory and Computation of Knapsack Functions," *Operations Research*, 14: 1045-1074 (1966).

Glover, F. "Surrogate Constraint," *Operations Research*, 16(4): 741-749 (1968).

Glover, F. "Surrogate Constraint Duality in Mathematical Programming," *Operations Research*, 23(3): 434-451 (1975).

Glover, F. "Heuristics for Integer Programming Using Surrogate Constraints," *Decision Sciences*, 8(1): 156-166 (1977).

Glover, F. "Future Paths for Integer Programming and Links to Artificial Intelligence," *Computers & Operations Research*, 13: 533-549 (1986).

Glover, F. "Optimization by Ghost Image Processes in Neural Networks," *Computers & Operations Research*, 21: 801-822 (1994).

Glover, F. and Kochenberger, G. "Critical Event Tabu Search for Multidimensional Knapsack Problems," in *Meta-Heuristics: Theory and Applications*, ed. I. H. Osman and J. P. Kelly, Kluwer Academic Publishers, 407-427 (1996).

Glover, F. and Laguna, M *Tabu Search*. Kluwer Academic Publishers (1997).

Godfrey, M.G., E.M. Roebuck, and A.J. Sherlock. *Concise Statistics*. Edward Arnold, 1988.

Hanafi, S. and Fréville, A. "An Efficient Tabu Search Approach for the 0 – 1 Multidimensional Knapsack Problem, " *European Journal of Operational* Research, 106: 659-675 (1998).

Hanafi, S., Fréville, A., and Abdellaoui, A. "Comparison of Heuristics for the 0 – 1 Multidimensional Knapsack Problem," in *Meta-Heuristics: Theory and Applications*, ed. I. H. Osman and J. P. Kelly, Kluwer Academic Publishers, 449-466 (1996).

Harder, R., Hill, R., and Moore, J. "A JAVA Universal Vehicle Router for Routing Unmanned Aerial Vehicles," *International Transactions in Operational Research*, 11(3): 259-276 (2004).

Hill, R. R. "Multivariate Sampling with Explicit Correlation Induction for Simulation and Optimization Studies," *Ph.D. Dissertation*, Department of Industrial, Welding, and Systems Engineering, The Ohio State University, Columbus, (1996).

Hill, R. R. "An Analytical Comparison of Optimization Problem Generation Methodologies," *Proceedings of the 1998 Winter Simulation Conference*, D.J. Medeiros, D.J., Watson, E.F., Carson, J.S., and Manivannan, M.S. (Editors), 609-615. Institute of Electrical and Electronics Engineers, Washington, DC (1998).

Hill, R. R. and C. Reilly. "The Effects of Coefficient Correlation Structure," *Management Science*, 46(2):302-317 (2000).

Hill, R. R. and C. Reilly. "The Effects of Coefficient Correlation Structure," *Proceedings of the 1994 Winter Simulation Conference*, J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila. (Editors), 332-339. Institute of Electrical and Electronics Engineers, Washington, DC (1994).

Hill, R., Cho, Y., and Moore, J. "Empirical Analysis of Legacy Heuristic using Simulation-Based Methods." *Proceedings of the 2003 Summer Computer Simulation Conference,* (2003a).

Hillier, F. S. "Efficient Heuristic Procedures for Integer Linear Programming with an Interior," *Operations Research*, 17(2): 600-637 (1969).

Hoff, A, Løkketangen, A, and Mittet, I. "Genetic Algorithms for 0/1 Multidimensional Knapsack Problems," *Working Paper,* Molde College, Britveien 2, 6400 Molde Norway (1996).

Holland, J. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ahn Arbor, MI, 1975.

Hooker, J. N. "An Empirical Science of Algorithms," *Operations Research*, 42(2): 201-212 (1994).

Iman, R., and Conover, W. "A Distribution-Free Approach to Inducing Rank Correlation Among Input Variables," *Commun. Statist.-Simula. Computa.*, 11(3): 311- 334 (1982).

Johnson, E., Kostreva, M. and Suhl, U. "Solving 0 – 1 Integer Programming Problems arising from Large Scale Planning Models," *Operations Research*, 33: 805-819 (1985).

Khuri, S., Bäck, T., and Heitkötter, J. "The Zero/One Multiple Knapsack Problem and Genetic Algorithms," *Proceedings ACM Symposium of Applied Computation*, *SAC '94* (1994).

Kirkpatrick, S., Gelatt, C., and Vecchi, M. "Optimization by Simulated Annealing," *Science*, 220: 671-680 (1983).

Klee, V. and Minty, J. "How Good is the Simplex Algorithm?," *Inequalities, III*, O. Shisha (ed.), Academic Press, New York, NY, 159-175 (1972).

Kochenberger, G., McCarl, B., and Wyman, F. "A Heuristic for General Integer Programming," *Decision Sciences*, 5(1): 36-44 (1974).

Lee, J. and Guignard, M. "An Approximate Algorithm for Multidimensional Zero-One Knapsack Problems – A Parametric Approach," *Management Science*, 34: 402-410 (1988).

Lin, G, and Rardin R. "Controlled Experimental Design for Statistical Comparison of Integer Programming Algorithms," *Management Science*, 25: 1250-1271 (1980).

Løkketangen, A. and Glover, F. "Solving Zero-One Mixed Integer Programming Problems Using Tabu Search," *European Journal of Operational Research*, 106: 624-658 (1998).

Lorie, J. and Savage, L. "Three Problems in Capital Rationing," *Journal of Business*, 28: 229-239 (1955).

Loulou, R., and Michaelides, E. "New Greedy Heuristics for the Multidimensional 0 – 1 Knapsack Problem," *Operations Research*, 27(6): 1101- 1114 (1979).

Magazine, M. and Chern, S. "A Note on Approximation Schemes for Multidimensional Knapsack Problems," *Mathematics of Operations Research*, 9: 244**-** 247 (1984).

Magazine, M. and Oguz, O. "A Heuristic Algorithm for the Multidimensional Zero-One Knapsack Problem," *European Journal of Operational Research*, 16: 319-326 (1984).

.

Manne, A. and Markowitz, H. "On the Solution of Discrete Programming Problems," *Econometrica*, 25: 84-110 (1957).

Mansini, R. and Speranza, M. "A Multidimensional Knapsack Model for the Asset-Backed Securitization," *Journal of Operational Research Society*, 53:822-832, (2002).

Martello, S. Psinger, D., and Toth, P. "Dynamic Programming and Strong Bounds for the 0 – 1 Knapsack Problem," *Management Science*, 45(3): 414-424 (1999).

Martello, S. and Toth, P. "A New Algorithm for the 0 – 1 Knapsack Problem," *Management Science*, 34(5): 633-644 (1988).

Martello, S. and Toth, P. "Upper Bounds and Algorithms for Hard 0 – 1 Knapsack Problems," *Operations Research*, 45(5): 768-777 (1997).

Martello, S. and Toth, P. "An Exact Algorithm for the Two-Constraint 0 – 1 Knapsack Problem," *Operations Research*, 51(5): 826-835 (2003).

Nemhauser, G. and Ullmann, Z. "Discrete Dynamic Programming and Capital Allocation," *Management Science*, 15: 494-505 (1969).

Nemhauser, G. and Wolsey L. *Integer and Combinatorial Optimization*. New York: John Wiley & Sons, Inc, 1988.

Oppen, J., Grüner, S., and Løkketangen, A. "A Tabu Search Based Heuristic for the 0/1 Multiconstrained Knapsack Problem," *Proceedings of the 2003 Norsk Informatikkonferanse* (2003).

Osorio, M., Glover, F and Hammer. P "Cutting and Surrogate Constraint Analysis for Improved Multidimensional Knapsack Solutions, " *Annals of Operations Research*, 117: 71-93 (2002).

Parker, R., and Rardin, R. “An Overview of Complexity Theory in Discrete Optimizations: Part I. Concepts,” *IIE Transactions*, 14(1): 3-10 (1982).

Peterson, C. “Computational Experience with Variants of the Balas Algorithm Applied to the Selection of Research and Development Projects,” *Management Science*, 13: 736-750 (1967).

Pirkul, H. “A Heuristic Solution Procedure for the Multiconstraint Zero-One Knapsack Problem,” *Naval Research Logistics*, 34(2): 161-172 (1987).

Plateau, G. and Elkihel, M. “A Hybrid Algorithm for the 0 – 1 Knapsack Problem” *Methods of Operations Research*, 49: 277-293 (1985).

Psinger, D. “A Minimal Algorithm for the 0 – 1 Knapsack Problem,” *Operations Research*, 45: 758-767 (1997).

Psinger, D. “Core Problems in Knapsack Algorithms,” *Operations Research*, 47(4): 570-575 (1999).

Raidl, G. “An Improved Genetic Algorithm for the Multiconstrained 0 – 1 Knapsack Problem,” In D. Fogel et al., editors, *Proceedings of the 5th IEEE International Conference on Evolutionary Computation*, 207-211 (1998).

Rardin, R. and Uzsoy, R. “Experimental Evaluation of Heuristic Optimization Algorithms: A Tutorial,” *Journal of Heuristics*, 7: 261-304 (2001).

Reeves, C. *Modern Heuristic Techniques for Combinatorial Problems*. McGraw-Hill Book Company Europe, 1995.

Reilly, C. “Properties of Synthetic Optimization Problems,” *Proceedings of the 1998 Winter Simulation Conference*, Medeiros, D.J., Watson, E.F., Carson, J.S., and Manivannan, M.S. (Editors), 621-627. Institute of Electrical and Electronics Engineers, Washington, DC (1998).

Rothkopf, M., Pekec, A., and Harstad, M. “Computationally Manageable Combinatorial Auctions,” *Rutgers University Technical Report*, Piscataway, NJ (1995).

Rousseeuw, P. and Molenberghs, G. “The Shape of Correlation Matrices,” *The American Statistician*, 48(4): 276-279 (1994).

Senju, S. and Toyoda, Y. “An Approach to Linear Programming with 0 – 1 Variables,” *Management Science*, 15(4): B196- B207 (1968).

Shih, W. "A Branch and Bound Method for the Multiconstraint Zero-One Knapsack Problems," *Journal of the Operations Research Society*, 30: 369-378 (1979).

Soyster, A., Lev, B., and Slivka, W. "Zero-One Programming with Many Variables and Few Constraints," *European Journal of Operational Research*, 2: 195-201 (1978).

Tammeraid, I., Majak, J., Pohjolainen, S., and Luodeslampi, T. *Linear Algebra*. Web Book: http://www.cs.ut.ee/~toomas_l/linalg/, 2004.

Thesen, A. "A Recursive Branch and Bound Algorithm for the Multidimensional Knapsack Problem," *Naval Research Logistics Quarterly*, 22: 341-353 (1975).

Thiel, J. and Voss, S. "Some Experiences on Solving Multiconstraint Zero-One Knapsack Problems with Genetic Algorithms," *INFOR*, 32(4): 226-242 (1994).

Toyoda, Y. "A Simplified Algorithm for Obtaining Approximate Solutions to Zero-One Programming Problems," *Management Science*, 21(12): 1417-1427 (1975).

Vasquez, M. and Hao, J. "A Hybrid Approach for the 0 – 1 Multidimensional Knapsack Problem ," *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJACAI-01),* 328-333 (2001).

Volgenant, A. and Zoon, J. "An Improved Heuristic for Multidimensional 0 – 1 Knapsack Problems," *Journal of the Operational Research Society,* 41: 963-970 (1990).

Weingartner, H. "Capital Budgeting of Interrelated Projects: Survey and Synthesis," *Management Science,* 12: 485-516 (1966).

Weingartner, H. and Ness, D. "Methods for the Solution of Multidimensional 0/1 Knapsack Problems," *Operations Research,* 15: 83-103 (1967).

Zanakis, S. "Heuristic 0 – 1 Linear Programming: An Experimental Comparison of Three Methods," *Management Science,* 24: 91-104 (1977).

Zionts, S. "Generalized Implicit Enumeration Bounds on Variables for Solving Linear Program with Zero-One Variables," *Naval Research Logistics Quarterly*, 19: 165-181 (1972).

Vıta

Major Yong Kun Cho was born in Seoul, Republic of Korea. He graduated from Nam-Kang High School in 1989. He then entered the Korea Military Academy. He received the Bachelor of Science degree in Civil Engineering in March 1993.

Upon graduation, he received the commission of 2nd Lieutenant, Army Infantry Officer. He was assigned to the 7th Division as a platoon leader. He led forty combat ready soldiers providing security against North Korean troops in the DMZ. Then he was assigned to ROK-US Combined Force Command, Seoul, Korea as Intelligence and Operation Officer. There, he conducted coordination between U.S command and ROK command providing cooperative operation for effective communication between U.S. Army and ROK Army. He then became a company commander in Reconnaissance Battalion, 11th Infantry Division. He commanded one hundred combat ready soldiers including three officers. He conducted many company field trainings and counter-terrorism operations. He entered the Graduate School of Engineering, Air Force Institute of Technology in August 2000 and received the Master of Science degree in Operations Research in March 2002. Upon graduation, he continued in the Ph.D. program at AFIT and achieved the Doctor of Philosophy in Operations Research in March 2005. He was inducted into a Tau Beta Pi (TBP-The National Engineering Honor Society) and an Omega Rho (The International Honor Society) while in AFIT.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
21-03-2005	**Doctoral Dissertation**	April 2002 – March 2005

4. TITLE AND SUBTITLE	
DEVELOPING NEW MULTIDIMENSIONAL KNAPSACK HEURISITCS BASED ON EMPIRICAL ANALYSIS OF LEGACY HEURISTICS	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	**5d. PROJECT NUMBER**
Cho, Yong Kun, Major, Republic of Korea Army	**5e. TASK NUMBER**
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) Hobson Way, Building 641 WPAFB OH 45433-7765	AFIT/DS/ENS/05-01
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**
AFSOR/NM, Attn: Dr. Neal Glassman, 4015 Wilson Boulevard Mail Room 713 Arlington, Virginia 22203-1954 Phone: (703) 696 -9548 E-mail: neal.glassman@afosr.af.mil	**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The multidimensional knapsack problem (MKP) has been used to model a variety of practical optimization and decision-making applications. Due to its combinatorial nature, heuristics are often employed to quickly find good solutions to MKPs. While there have been a variety of heuristics proposed for the MKP, and a plethora of empirical studies comparing the performance of these heuristics, little has been done to garner a deeper understanding of heuristic performance as a function of problem structure. This dissertation presents a research methodology, empirical and theoretical results explicitly aimed at gaining a deeper understanding of heuristic procedural performance as a function of test problem characteristics. This work first employs an available, robust set of two-dimensional knapsack problems in an empirical study to garner performance insights. These performance insights are tested against a larger set of problems, five-dimensional knapsack problems specifically generated for empirical testing purposes. The performance insights are found to hold in the higher dimensions. These insights are used to formulate and test a suite of three new greedy heuristics for the MKP, each improving upon its successor. These heuristics are found to outperform available legacy heuristics across a complete spectrum of test problems. Problem reduction heuristics are examined and the subsequent performance insights garnered are used to derive a new problem reduction heuristic, which is then further extended to employ a local improvement phase. These problem reduction heuristics are also found to outperform currently available approaches. Available problem test sets are shown lacking along multiple dimensions of importance for viable empirical testing. A new problem generation methodology is developed and shown to overcome the current limitations in available problem test sets. This problem generation methodology is used to generate a new set of empirical test problems specifically designed for competitive computational tests. This new test set is shown to stress existing heuristics; not only does the computational time required by these legacy heuristics increase with problem size, but solution quality is found to decrease with problem size. However, the solution quality obtained by the suite of heuristics developed in this dissertation are shown to be unaffected by problem size thereby providing a level of robust solution quality not previously seen in heuristic development for the MKP. This research demonstrates that the test problems can have a profound, and sometimes misleading, impact on the general insights gained via empirical testing, provides six new quality heuristics, and two new robust sets of test problems, one focused on empirical testing, the other focused on competitive testing.

15. SUBJECT TERMS

Multidimensional Knapsack Problem, Heuristics, Optimization, Experimental Design

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			James, T. Moore, Ph.D. (AFIT/ENS)
U	**U**	**U**	UU	254	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636, ext 4528; e-mail: james.moore@afit.edu