Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers & Industrial Engineering 48 (2005) 83–96
computers & industrial engineering
www.elsevier.com/locate/dsw

Meta-RaPS approach for the 0-1 Multidimensional Knapsack Problem

Reinaldo J. Moraga[a,*], Gail W. DePuy[b], Gary E. Whitehouse[c]

[a]Department of Industrial Engineering, Universidad del Bío Bío, Avenida Collao 1202, Casilla 5C, Concepción, Chile
[b]Department of Industrial Engineering, University of Louisville, Louisville, KY 40292, USA
[c]Industrial Engineering and Management Systems Department, University of Central Florida, Orlando, FL 32816-2450, USA

Received 3 October 2003; accepted 7 February 2004
Available online 2 November 2004

Abstract

A promising solution approach called Meta-RaPS is presented for the 0-1 Multidimensional Knapsack Problem (0-1 MKP). Meta-RaPS constructs feasible solutions at each iteration through the utilization of a priority rule used in a randomized fashion. Four different greedy priority rules are implemented within Meta-RaPS and compared. These rules differ in the way the corresponding pseudo-utility ratios for ranking variables are computed. In addition, two simple local search techniques within Meta-RaPS' improvement stage are implemented. The Meta-RaPS approach is tested on several established test sets, and the solution values are compared to both the optimal values and the results of other 0-1 MKP solution techniques. The Meta-RaPS approach outperforms many other solution methodologies in terms of differences from the optimal value and number of optimal solutions obtained. The advantage of the Meta-RaPS approach is that it is easy to understand and easy to implement, and it achieves good results.

Keywords: Meta-heuristics; Optimization; Greedy algorithm; 0-1 Multidimensional Knapsack Problem

1. Introduction

Significant attention has been lately focused on the 0-1 Multidimensional Knapsack Problem (0-1 MKP), which is a well-known NP-hard combinatorial problem. The importance of the 0-1 MKP arises from both real-world and theoretical standpoints. The 0-1 MKP embraces many practical

* Corresponding author. Tel.: +56-41-731701, fax: +56-41-731021.
E-mail addresses: rmoraga@ubiobio.cl (R.J. Moraga), depuy@louisville.edu (G.W. DePuy), whitehse@mail.ucf.edu (G.E. Whitehouse).

0360-8352/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2004.02.008

applications, some of them include cutting stock problems (Gilmore & Gomory, 1966), loading problems (Bellman, 1957; Shih, 1979), delivery of groceries in vehicles with multiple compartments (Chajakis & Guignard, 1992), project selection (Kyparisis, Gupta, & Ip, 1996; Petersen, 1967), capital budgeting (Weingartner, 1967), and allocating processors and data in distributed computer systems (Gavish & Pirkul, 1982). In addition, the set covering and multicovering problems can be reformulated as an equivalent 0-1 MKP through variable complementing (Bertsimas & Demir, 2002; Hochbaum, 1996) and the vehicle routing problem can be reduced to the 0-1 knapsack problem for some special conditions (Eilon, Watson-Gabdy, & Christofides, 1971).

The simple knapsack problem seeks to determine which of n given items of merchandise to place and carry in a knapsack for sale in a far-away market such that the total profit is maximized while the total weight does not exceed a given limit. An extensive treatment of the simple knapsack problem is provided by Martello and Toth (1990). The 0-1 MKP involves a knapsack with m constraints with restrictions $b_1, b_2, b_3, \ldots, b_m$ and n items, each of which can earn a profit c_i. The ith item has a coefficient of a_{ij} when it is considered for the jth constraint. Typical constraints might include total weight, total volume, number of units, etc. In mathematical terms, the 0-1 MKP can be formulated as follows:

$$\text{Maximize } \sum_{i=1}^{n} c_i x_i \tag{1}$$

$$\text{Subject to } \sum_{i=1}^{n} a_{ij} x_i \leq b_j \quad \forall\, j = 1, \ldots, m \tag{2}$$

$$x_i \in \{0, 1\} \quad \forall\, i = 1, \ldots, n \tag{3}$$

$$\text{where } x_i = \left\{ \begin{array}{ll} 1 & \text{if item } i \text{ included in knapsack} \\ 0 & \text{otherwise} \end{array} \right\}.$$

Several solution approaches are proposed in the literature for the 0-1 MKP. Exact algorithms based on a branch and bound technique are proposed by Gavish and Pirkul (1985), Osorio, Glover, and Hammer (2003), and Shih (1979). Some exact algorithms based on the use of dynamic programming methods have been also reported (Gilmore & Gomory, 1966; Weingartner & Ness, 1967) but with limited success for modest-size problems. Bertsimas and Demir (2002) introduced an approximate dynamic programming approach, which seems to be promising when compared with other methods.

Heuristic applications to the 0-1 MKP vary from simple ranking lists for constructing a solution to meta-heuristic based approaches. Greedy ranking heuristics are proposed by Eilon et al. (1971), Loulou and Michaelides (1979), Senyu and Toyoda (1968), and Toyoda (1975). Balas and Martin (1980), Chu and Beasley (1998), Greenberg and Pierskalla (1970), and Lee and Guignard (1988) develop heuristics based on the linear programming relaxation of the integer programming 0-1 MKP.

With respect to the use of meta-heuristics, the solution method used in this paper, Hanafi, Fréville, and El Abdellaoui (1996) developed a simple multistage algorithm with a threshold accepting search (SMA/TA) for the 0-1 MKP, which consisted of generating a set of random solutions that are used as

starting solutions for a neighborhood search. If any starting solution violated at least one constraint, a repair mechanism was utilized. In addition, they proposed a tabu search variant to deal with infeasibility (IFTS/HFE). Fréville and Plateau (1994) proposed an efficient preprocessing procedure for the 0-1 MKP. Their greedy method, called AGNES, is composed of three concepts first introduced by Glover (1965, 1977): surrogate relaxation, oscillating assignment, and strongly determined variables. Drexel (1988) considered a simulated annealing algorithm for the 0-1 MKP. Drexel's (1988) approach was implemented by Dammeyer and Voss (1993) and compared to their tabu search strategy that used the reverse elimination method (Glover, 1990) for a dynamic updating of the tabu list. Tabu search compared favorably to the simulated annealing approach. Glover and Kochenberger (1996) and Hanafi and Fréville (1998) propose a tabu search based method for the 0-1 MKP with a flexible memory structure that was updated at critical events. Also genetic algorithms were proposed by Chu and Beasley (1998), Haul and Voss (1997), and Khuri, Bäck, and Heitkötter (1994). It is meaningful to note that Chu and Beasley (1998) provide a very good literature survey of the 0-1 MKP and pose their own test suite for large size problems available through the OR-Library (Beasley, 1990).

This article presents the application of a promising meta-heuristic approach called Meta-RaPS (Meta-heuristic for Randomized Priority Search) to the 0-1 MKP as an extension of prior work (Moraga, 2002; Moraga, DePuy, & Whitehouse, 2003). Meta-RaPS is a generic, high-level strategy used to construct and improve feasible solutions through the utilization of simple heuristic rules used in a randomized fashion. As with other meta-heuristics, the randomness represents a device to avoid getting trapped at a local optimal solution.

In general, a construction heuristic builds a solution by adding feasible items (of merchandise) to the current solution. The item with the best priority value is added to the current solution. Meta-RaPS modifies the way a general construction heuristic chooses the next item to add to the solution by occasionally choosing an item that does not have the best priority value. Once a solution has been constructed, Meta-RaPS may proceed to further improve it through neighborhood search techniques. Meta-RaPS controls the construction and search using four parameters; number of iterations (I), %priority ($\%p$), %restriction ($\%r$), and %improvement ($\%i$).

The number of iterations determines the number of feasible solutions constructed. The %priority parameter is used to determine the percentage of time the next item added to the solution has the best priority value. The remaining time (100% − %priority) the next item added to the solution is randomly chosen from those feasible items whose priority values are within %restriction below the best priority value. Proceeding in this way, Meta-RaPS guarantees a diversification process when a solution is being constructed while keeping the quality produced by the priority rule. In addition, a solution improvement algorithm (neighborhood search) can be included in Meta-RaPS using the %improvement parameter. The improvement heuristic (neighborhood search) is performed if an iteration's solution value is within %improvement of the best unimproved solution value found so far. The best solution from all iterations is reported.

This paper is organized as follows. Section 2 discusses the construction rules and improvement techniques that will be used by Meta-RaPS for the 0-1 MKP. In Section 3, computational results are reported as well as comparisons to other approaches. Also presented in Section 3 is a method for determining good values for the four parameters. Concluding remarks, including worthy future research topics, are given in Section 4.

2. Meta-RaPS application to the 0-1 MKP

As mentioned in the Section 1, Meta-RaPS is a master strategy that uses both construction and improvement heuristics to generate high quality solutions. In this paper, Meta-RaPS is applied to four 0-1 MKP construction heuristics described in detail in this section: Simplest Greedy Rule (SGR), Dual-Relaxation Greedy Rule (DRGR), Dynamic Greedy Rule (DGR), and Loulou & Michaelides Greedy Rule (LMGR). Each of these construction heuristics is a greedy rule that begins by calculating a pseudo-utility ratio for each item, i.e. $\alpha_i = c_i/w_i$; where w_i is the penalty factor for item i (Loulou & Michaelides, 1979). Next, the items are ordered by non-increasing magnitude of their pseudo-utility ratios and put in a priority list. The ordered items are then considered for inclusion in the current solution beginning at the top of the priority list. The ith item is assigned to the current solution if no constraint is violated; otherwise, the item is discarded.

Meta-RaPS modifies the way a construction heuristic chooses the next item to add to the solution by occasionally choosing a feasible item that does not have the largest pseudo-utility ratio. In general, Meta-RaPS is applied to a single greedy construction heuristic (either SGR, or DRGR, or DGR, or LMGR, or another greedy MKP construction heuristic not discussed in this paper) through the use of four user-defined parameters; %priority, %restriction, %improvement, number of iterations. In Meta-RaPS MKP, the %priority parameter defines the percentage of time the next item added to the knapsack has the largest pseudo-utility ratio. The remaining time (i.e. 100% − %priority) the next item is randomly chosen from those items whose pseudo-utility ratios are within %restriction below the largest pseudo-utility ratio. Specifically, if parameters of 20% priority and 40% restriction are used, then 20% of the time the item with the largest pseudo-utility ratio is added to the knapsack and the remaining 80% of the time the next item is randomly selected from a list of those feasible items whose pseudo-utility ratios are greater than 0.40× (largest pseudo-utility ratio). This method of random selection is thought to decrease the possibility of being trapped in local optima, and helps the user to find an answer closer to the global optimum. Next, an improvement heuristic is implemented on those constructed solutions that are within %improvement below the best unimproved solutions. The best solution of overall iterations is reported.

Meta-RaPS is the result of research conducted on the application of a modified COMSOAL approach to several combinatorial problems. Computer Method of Sequencing Operations for Assembly Lines (COMSOAL) is a computer heuristic originally reported as a solution approach to the assembly line balancing problem (Arcus, 1966). Despite the fact that Meta-RaPS conserves in essence Arcus's underlying original COMSOAL idea, in practice, the methods differ considerably. Meta-RaPS has been successfully applied to a number of combinatorial problems such as the Resource Constrained Project Scheduling Problem (DePuy & Whitehouse, 2001), the Traveling Salesman Problem (DePuy, Moraga, & Whitehouse, 2004; Moraga, 2002), the Bin Packing Problem (DePuy, Whitehouse, & Moraga, 2003), and the Vehicle Routing Problem (Moraga, 2002).

Meta-RaPS can be considered a general form of COMSOAL. Using Meta-RaPS parameters of 0%priority, an infinitely large %restriction, and 0%improvement will mimic COMSOAL. Meta-RaPS is also the general form of another meta-heuristic called GRASP (Greedy Randomized Adaptive Search Procedure; Feo & Resende, 1995). GRASP constructs solutions by introducing randomness to a greedy construction heuristic through the use of a %restriction parameter in the same way as Meta-RaPS, but it does not give any probabilistic priority to the alternative considered the best by the greedy construction heuristic. GRASP also includes a local search improvement heuristic applied to all constructed solutions. GRASP has been applied to a variety of problems, however, no research solving the 0-1 MKP problems

using GRASP has been found to date. Using Meta-RaPS parameters of 0%priority and 100%improvement will imitate GRASP. Meta-RaPS offers greater flexibility over COMSOAL and GRASP in that it allows user-defined settings for three parameters (%priority, %restriction, and %improvement). In addition, GRASP uses g reedy algorithms while Meta-RaPS has been used with both greedy algorithms and non-greedy or look-ahead priority rules (DePuy et al., 2003, 2004). As will be demonstrated in the following sections, the Meta-RaPS' parameter settings used to find the best solutions are settings other than those that imitate COMSOAL or GRASP. Therefore, the extra flexibility afforded by the general form of Meta-RaPS seems beneficial.

2.1. MKP construction heuristics

As mentioned previously, four greedy 0-1 MKP construction rules are implemented separately within Meta-RaPS. These four schemes distinguish themselves in the way the penalty factor is computed and are briefly described as follows.

2.1.1. Simplest Greedy Rule (SGR)

Eilon et al. (1971) suggest this rule, which transforms the whole family of original constraints into one new equivalent constraint. The penalty factor for the ith item is given by the following expression:

$$w_i = \sum_{j=1}^{m} \left(\frac{a_{ij}}{b_j} \right) \tag{4}$$

DePuy et al. (2003) and Moraga (2002) report the use of this rule in Meta-RaPS.

2.1.2. Dual-Relaxation Greedy Rule (DRGR)

Chu and Beasley (1998) use this rule as a repair operator for their approach based on genetic algorithms. Unlike the SGR, this rule multiplies each constraint by its corresponding dual multiplier λ_j, where each λ_j is the shadow price of the jth constraint in the LP-relaxation of the 0-1 MKP. The penalty factor for the ith item is then given by:

$$w_i = \sum_{j=1}^{m} a_{ij} \lambda_j \tag{5}$$

2.1.3. Dynamic Greedy Rule (DGR)

Moraga (2002) proposes this rule of dynamic type. The penalty factors for each item change their magnitude as items are assigned to the current solution. The penalty factor is computed by dividing each constraint by its corresponding available amount of jth resource, $(b_j - \mathrm{CW}_j)$; where CW_j is the amount of jth resource consumed by the items assigned so far. The penalty factor for the ith item is given by the following expression:

$$w_i = \sum_{j=1}^{m} \left(\frac{a_{ij}}{b_j - \mathrm{CW}_j} \right) \tag{6}$$

2.1.4. Loulou and Michaelides Greedy Rule (LMGR)

Loulou and Michaelides' (1979) greedy heuristic rule is based on Toyoda's work (1975). Their procedure starts with a zero solution and systematically adds a new item to the current solution by choosing the item with highest pseudo-utility ratio among those candidate variables that satisfy all constraints. The LMGR consists of calculating the penalty factor w_i by considering the worst effect that the ith item would have on the three following quantities of the jth resource (Loulou & Michaelides, 1978):

- $\mathrm{CW}_j + a_{ij}$: total consumption of resource j if the item i is added to the current solution.
- $\{b_j - (\mathrm{CW}_j + a_{ij})\}$: amount of resource j remaining if item i is added to the current solution.
- $\sum_{k \in K}(a_{kj} - a_{ij})$: future potential demand for resource j if item i is added to the current solution (K is the set of candidate items).

The choice for w_i is then:

$$w_i = \max_{j=1,\ldots,m}\left\{\frac{(\mathrm{CW}_j + a_{ij})\left(\sum_{k \in K}(a_{kj} - a_{ij})\right)}{b_j - \mathrm{CW}_j - a_{ij}}\right\} \tag{7}$$

It should be noted that the SGR and DRGR construction heuristics are static in nature while DGR and LMGR are dynamic in that the penalty factors for each item change in magnitude as items are assigned to the current solution.

2.2. MKP improvement heuristics

The improvement stage implemented within Meta-RaPS MKP basically consists of two neighborhood search techniques: insertion and exchanging. First, one of the non-assigned items is randomly selected and tried to insert into the current solution without violating feasibility. If this move is possible, the solution will actually improve and the procedure goes to the next non-assigned item. Second, if the insertion is not possible, then the current non-assigned item starts being exchanged for an assigned item assigned that is randomly selected. If feasibility is not violated and the solution improves, the move is kept. Otherwise, the move is discarded, and a next assigned item is selected randomly. The Meta-RaPS MKP approach is outlined in Fig. 1.

3. Computational study

Meta-RaPS MKP, as described in Section 2, was coded in C++, and runs were made on a Pentium IV 1.6 GHz PC. This results section is presented in four parts. First, a description of the method used to select the Meta-RaPS' parameters is presented. Then, the benefit of incorporating randomness in a simple greedy construction heuristic (i.e. the basic premise of the Meta-RaPS approach) is investigated. A standard library of 56 small 0-1 MKP test problems published in the literature and available from

```
Load instance; Input parameters, I, %p, %r, %i; best_unimpr_z = best_z =-∞;
While num_iter ≤ I
    S={x1, x2,…,xn}; X0={∅}; X1={∅}; z=0;
    For v=1,…, n
        p = rnd(1..100);
        If p ≤ %p
            find k such that, α_k = max_{t∈S}(c_t / w_t);
            If x_k is feasible, add x_k to X1; update z; Else add x_k to X0; End_if
            delete x_k from S;
        Else
            find α = max_{t∈S}(c_t / w_t);
            pick k randomly from {t∈ S | c_t/w_t ≥ α[1-(%r/100)]};
            If x_k is feasible, add x_k to X1; update z; Else add x_k to X0; End_if
            delete x_k from S;
        End_if
    End_for
    If z > best_unimpr_z, best_unimpr_z=z; End_if
    If z > [1-(%i/100)]best_unimpr_z,
        null_elements=X0;
        While null_elements≠{∅}
            pick x_h from null_elements randomly; z'=z;
            If x_h is feasible, INSERT x_h into the solution; update z';
            Else
                one_elements=X1; search=true;
                While one_elements≠{∅} or search is true
                    pick k from one_elements randomly;
                    If exchanging(x_h, x_k) is feasible and z' improves,
                            EXCHANGE(x_h, x_k); update z'; search = false;
                    End_if
                    delete x_k from one_elements;
                End_while
            End_if
            delete x_h from null_elements;
            keep best_z' and solution;
        End_while
        If best_z'>z, z= best_z'; update(X0, X1); End_if
    End_if
    If z>best_z, best_z=z; X={X0∪X1}; End_if
    num_iter= num_iter+1;
End_while
Report best_z and X;
```

Fig. 1. Meta-RaPS' pseudocode for 0-1 MKP.

the OR-Library (Beasley, 1990) and the MP-TESTDATA (Skorobohatyj, 2002) is used to evaluate the Meta-RaPS approach using the four greedy construction heuristics described in Section 2.1. Next, the benefit of including an improvement heuristic in the Meta-RaPS approach is discussed. Finally, a set of 270 large problems developed by Chu and Beasley (1998), which are also available at the OR-Library (Beasley, 1990), are evaluated using the best greedy construction heuristic selected based on the results of the earlier experiment. For both test problem sets, the results are compared to both the optimal solutions and the results obtained with other techniques.

3.1. Parameter setting

Parameters in Meta-RaPS can be set either arbitrarily or by using a more systematic method. In this case, the following general parameter setting framework (Moraga, 2002) was used:

Step 1. Select a representative subset of problems to analyze from the entire collection of problems. In this case, a subset of 10% of the entire problem test set was randomly selected.
Step 2. Select the parameter domain over which each parameter will be varied. The parameter domain was varied over the whole range (0–100%) for $\%p$ and $\%r$, and over a 0–20% range for $\%i$. An increment size of 10% was used for parameter $\%p$. While in the case of parameter $\%r$, increment sizes of 10% were used for small size problems and increment sizes of 1% were used for large size problems. For parameter $\%i$, increment sizes of 5% were used for small size problems and increment sizes of 0.1% were used for large size problems. The number of iterations, I, was fixed either to 10,000, 5000, or 1000 for small, medium and large size problems, respectively.
Step 3. For each problem in the subset, run Meta-RaPS MKP using an appropriate technique for setting parameters. In this step, the technique consisted of sequentially varying each parameter over its domain until a value is found while keeping others at fixed values. First, the $\%r$ parameter is varied while fixing $\%p$ and $\%i$ to zero. The $\%r$ parameter associated with the best solution value will be used. Second, the $\%p$ parameter is varied while maintaining $\%r$ at its setting and $\%i$ at zero. Finally, the $\%i$ parameter is varied while keeping $\%p$ and $\%r$ at their settings. The iterations number I was chosen by trading off quality solutions versus runtimes.
Step 4. Use the overall best parameter settings obtained in Step 3 for the entire collection of problems.

It should be noted that the parameter selection methodology presented in this section may not produce the optimal combination of parameter values. The intention is to provide a systematic method of selecting good parameters.

3.2. Results for the small problems

The 56 standard problems used in this experiment are considered small real-world problems consisting of $m = 2$–30 and $n = 6$–105 with known optimal values. A subset of six-problems (FLEI, HP2, PETERSEN6, WEING5, WEISH9, and WEISH21) were randomly selected to determine parameter settings as per the method described in Section 3.1. Based on this subset, parameter settings of $\%p = 30$, $\%r = 50$, and $I = 10{,}000$ were chosen for the entire test set of 56 0-1 MKP problems. Results are shown in Table 1 in terms of the number of optimal solutions reached, deviation percentage from the optimal solution, and run times for each of the four greedy construction heuristics.

Table 1
Meta-RaPS MKP results for 56 0-1 MKP problems

Solution method	No. of optimal solutions	Maximum % deviation from optimal	Average % deviation from optimal	Run times (s/prob)
SGR	4/56	11.331	3.711	0
DRGR	8/56	19.178	2.301	0
DGR	8/56	10.005	1.359	0.01
LMGR	8/56	15.335	1.739	0.01
Meta-RaPS SGR	36/56	2.384	0.179	7
Meta-RaPS DRGR	44/56	1.474	0.102	8
Meta-RaPS DGR	50/56	0.795	0.045	100
Meta-RaPS LMGR	44/56	1.474	0.079	120

As can be seen in Table 1, the inclusion of randomness (i.e. the Meta-RaPS approach) always outperformed the greedy construction heuristic. It can also be seen that the Meta-RaPS DGR construction heuristic dominates the other constructions heuristics.

3.3. Meta-RaPS approach with improvement heuristic

Now that the benefits of including randomness in a simple construction priority rule have been demonstrated, the task at hand becomes solution improvement. In addition to the three parameters previously discussed (I, $\%p$, and $\%r$), Meta-RaPS will now use the $\%i$ parameter for choosing whether or not to improve the current solution. If a current solution value before improvement is within %improvement ($\%i$) of the best unimproved solution value found so far (see Fig. 1), the improvement heuristic presented in Section 2.2 will be performed. For example, if an improvement parameter of 20% is used, the improvement heuristic will be applied to any unimproved solution whose value is greater than $(0.8)\times$ (best unimproved solution value to date). The idea behind this strategy is that, in general, good unimproved solutions lead to good improved solutions (DePuy et al., 2004).

Four levels of improvement, 5, 10, 15, and 20%, were considered for the six-problem subset used for parameter setting. It was found that results produced using 15 and 20% resulted in the best solution values. Therefore, a parameter of $\%i = 15$ was used for the entire collection of 56 0-1 MKP and results are summarized in Table 2. Meta-RaPS DGR outperforms the other methods finding 55 of the 56 optimal solutions. PETERSEN7 is the only problem for which Meta-RaPS DGR did not find the optimal solution in the reported 10,000 iterations, although the optimal solution was found for this problem during a subsequent run. Runtimes are expressed in seconds per problem (i.e. total runtime for 10,000 iterations).

Table 2
Meta-RaPS MKP results for 56 0-1 MKP problems with 15% improvement

Solution method	No. of optimal solutions	Average % deviation from optimal	Run times (s/prob)
Meta-RaPS SGR	38/56	0.105	12
Meta-RaPS DRGR	52/56	0.029	13
Meta-RaPS DGR	55/56	0.003	115
Meta-RaPS LMGR	50/56	0.042	138

Table 3
Comparison of Meta-RaPS to other MKP solution techniques for test problems

Solution method	Optimal solutions	Average % deviation from optimal
Meta-RaPS DGR	55/56	0.003
GRASP	52/56	0.023
SMA/TA (Hanafi et al., 1996)	39/54	0.080
AGNES (Fréville & Plateau, 1994 as reported by Hanafi et al., 1996)	52/54	0.020
Simulated annealing DETEXC (Drexel, 1988)	7/57	1.739
Simulated annealing PROEXC (Drexel, 1988)	23/57	0.239
Simulated annealing (Drexel, 1988 as implemented by Dammeyer & Voss, 1993)	31/57	0.328
Tabu search REM (Dammeyer & Voss, 1993)	40/57	0.126
Tabu search STM (Dammeyer & Voss, 1993)	39/57	0.130
Tabu search L+STM (Dammeyer & Voss, 1993)	44/57	0.101
Tabu search (Glover & Kochenberger, 1996)	57/57	0.000
Tabu search (Lokketangen & Glover, 1998)	37/54	0.003
Tabu search IFTS/HFE (Hanafi et al., 1996)	54/54	0.000
Genetic algorithm (Chu & Beasley, 1998)	55/55	0.000
Fix+cut based method (Osorio et al., 2003)	55/55	0.000

Based on these findings, only the Meta-RaPS DGR approach will be used throughout the remainder of this paper.

In addition, Meta-RaPS results were contrasted to several other well-known 0-1 MKP solution techniques used to solve the same standard problems, as shown in Table 3. The solution quality of Meta-RaPS compares favorably with that of other authors. Meta-RaPS outperforms all of Simulated Annealing implementations, GRASP, and most tabu search versions with exception of the one proposed by Glover and Kochenberger (1996). As indicated in Section 2, no published research could be found using GRASP to solve the 0-1 MKP, therefore, the GRASP results shown in Table 3 were simulated by the authors through Meta-RaPS DGR with $\%p=0$, $\%r=50$ and $\%i=100$. However, Meta-RaPS is slightly outperformed by Chu and Beasley's (1998) GA approach. Notice that some researchers used 57 test problems while other researchers used fewer test problems; this is because available libraries are not complete. In this paper, a collection of 56 problems was available from libraries from Beasley (1990) and Skorobohatyj (2002), there is a missing problem that was not possible to find. At this point, it is important to stress that Meta-RaPS DGR uses a very simple and straightforward construction heuristic making Meta-RaPS DGR attractive to industry practitioners. The GA technique proposed by Chu and Beasley (1998) uses the dual variables of the 0-1 MKP LP-relaxation as the basis for its repair operator, which may complicate the implementation of this approach in practice.

3.4. Results for large problems

Chu and Beasley (1998) report that their GA algorithm solves all instances from the test set used in Section 3.3. Thus, they created their own test set of larger and more difficult 0-1 MKP problems. Instances were generated using the procedure suggested by Fréville and Plateau (1994). The number of constraints m was set to 5, 10 and 30, and the number of variables n was set to 100, 250 and 500. A total of 270 problem were generated; 30 problems per each $m-n$ combination. Meta-RaPS DGR was applied

Table 4
Meta-RaPS' parameter settings for large problems

M	*n*	Number of instances	Parameter settings			
			I	*%p*	*%r*	*%I*
5	100	30	10,000	10	10	2
	250	30	5000	80	5	0.8
	500	30	1000	80	2	0.3
10	100	30	10,000	10	10	2
	250	30	5000	80	3	0.8
	500	30	1000	60	2	0.3
30	100	30	10,000	10	10	2
	250	30	5000	20	1	0.8
	500	30	1000	60	2	0.3

using the parameter settings shown in Table 4. These parameter settings were found using the method described in Section 3.1. As the problem size (i.e. *n*/*m* ratio) increases, the number of iterations was decreased in order to achieve a balance between quality solutions and run times.

Using the 270 large problem test set, Table 5 compares the results of Meta-RaPS DGR to the genetic algorithm methods of Chu and Beasley (1998) and Haul and Voss (1997) denoted as GA-CB and GA-HV, respectively, the approaches of Magazine and Oguz (1984), Pirkul (1987), and Volgenant and Zoon (1990) denoted as MKP-P, MKP-MO, MKP-VZ, respectively, and the approximate dynamic programming method of Bertsimas and Demir (2002) denoted as ADP. Because the optimum solution values are not known for these large problems, Table 5 shows the average percent deviation from the optimum values of the LP-relaxations which serve as upper bounds on the optimal objective values.

Meta-RaPS DGR dominates the other solution techniques except Chu and Beasley's (1998) genetic algorithm in terms of overall percent deviation. For the largest problems ($n=30$, $m=250$ and 500) Meta-RaPS DGR does not perform as well as Chu and Beasley's (1998) genetic algorithm or Bertsimas and Demir's (2002) approximate dynamic programming. However, it should be reiterated that

Table 5
Comparison of Meta-RaPS DGR to other solution techniques for 270 large test problems

Problem		Average % deviation from LP optimum						
M	*n*	MR DGR	GA-CB	GA-HV	MKP-P	MK-MO	MKP-VZ	ADP
5	100	0.60	0.59	0.72	0.95	8.49	7.63	N/A
5	250	0.17	0.14	0.36	0.31	5.14	4.61	N/A
5	500	0.09	0.05	0.34	0.12	3.40	3.02	N/A
10	100	1.17	0.94	1.26	2.12	10.79	10.65	N/A
10	250	0.45	0.30	0.74	0.66	7.66	6.74	N/A
10	500	0.20	0.14	0.64	0.29	6.05	4.99	N/A
30	100	2.23	1.69	2.14	4.85	11.93	11.11	N/A
30	250	1.38	0.68	1.36	2.02	8.89	7.81	0.97
30	500	0.82	0.35	1.20	1.03	6.89	6.28	0.52
Overall		0.77	0.53	0.93	1.34	7.12	6.47	0.74

Meta-RaPS is a general solution approach that can be applied to many different combinatorial optimization problems and as such Meta-RaPS DGR compares favorably to both other general solution techniques (simulated annealing, tabu search, genetic algorithms) and heuristics developed specifically for the 0-1 MKP. Meta-RaPS DGR run times ranged from 7 to 35 min per problem. Average execution times for Chu and Beasley's (1998) genetic algorithm ranged from 6 to 65 min on a Silicon Graphics Indigo workstation. Haul and Voss (1997) report that their algorithm takes a very long time to solve large instances, in some cases more than 4 h on Intel Pentium 100 MHz PC. On the other hand, Bertsimas and Demir's (2002) report an average computation time of 87.06 s on a Dell Precision 410 machine. While, it is difficult to compare run times because different computer platforms are used by different researchers, it seems that Bertsimas and Demir's (2002) method outperforms the other methods in terms of run time. There is no runtime information available on the other approaches.

4. Summary and conclusions

Based on the computational evidence presented in Section 3, Meta-RaPS appears to be a viable solution approach for the 0-1 MKP. Four greedy 0-1 MKP construction heuristics were incorporated in Meta-RaPS with the Dynamic Greedy Rule (DGR) offering the best results. The Meta-RaPS DGR heuristic developed in this paper achieved good results when compared to both the optimal solution and other 0-1 MKP solution techniques such as simulated annealing, tabu search, genetic algorithms, and 0-1 MKP heuristics. The Meta-RaPS method should be particularly attractive to industrial practitioners as it is easy to comprehend and easy to put into practice, and produces good results in a reasonable amount of time. Many existing 0-1 MKP methods are complex and require a mature understanding of mathematical programming and/or computer programming.

It should be noted that the performance of Meta-RaPS DGR was not quite as good as that of Chu and Beasley's (1998) genetic algorithm or Bertsimas and Demir's (2002) approximate dynamic programming for the largest problem sizes. Further investigation can be done to improve the solution quality for these large problems. In addition, there is opportunity to enhance the simple local search improvement heuristics implemented in Meta-RaPS DGR and to possibly achieve better results.

Acknowledgements

This research has been partially supported by R&D Department grant 024409 3/R from Universidad del Bío Bío.

References

Arcus, A. L. (1966). COMSOAL: A computer method of sequencing operations for assembly lines. *International Journal of Production Research*, *4*, 259–277.

Balas, E., & Martin, C. H. (1980). Pivot and complement—a heuristic for 0-1 programming. *Management Science*, *26*, 86–96.

Beasley, J. E. (1990). OR-Library: Distributing test problems by electronic mail. *Journal of the Operational Journal Society*, *41*, 170–181. http://www.ms.ic.ac.uk/info.html.

Bellman, R. (1957). *Dynamic programming*. Princeton, NJ: Princeton University Press.

Bertsimas, D., & Demir, R. (2002). An approximate dynamic programming approach to multidimensional knapsack problem. *Management Science*, *48*(4), 550–565.

Chajakis, E., Guignard, M. (1992). A model for delivery of groceries in vehicle with multiple compartments and Lagrangean approximation schemes. In *Proceedings of Congreso Latino Ibero-Americano de Investigación de Operaciones e Ingeniería de Sistemas* 1992, Mexico City.

Chu, P. C., & Beasley, J. E. (1998). A genetic algorithm for the multidimensional knapsack problem. *Journal of Heuristics*, *4*, 63–86.

Dammeyer, F., & Voss, S. (1993). Dynamic tabu list management using the reverse elimination method. *Annals of Operations Research*, *41*, 31–46.

DePuy, G., & Whitehouse, G. (2001). A simple and effective heuristic for the multiple resource allocation problem. *International Journal of Production Research*, *32*(4), 24–31.

DePuy, G., Whitehouse, G., & Moraga, R. (2003). Using the Meta-RaPs approach to solve combinatorial problems. *Computers and Industrial Engineering* (Under review).

DePuy, G. W., Moraga, R. J., & Whitehouse, G. E. (2004). Meta-RaPS: A simple and effective approach for solving the traveling salesman problem. *Transportation Research Part E: Logistics and Transportation Review* (Under review).

Drexel, A. (1988). A simulated annealing approach to the multiconstraint zero-one knapsack problem. *Computing*, *40*, 1–8.

Eilon, S., Watson-Gabdy, C., & Christofides, N. (1971). *Distribution management: Mathematical modeling and practical analysis*. New York: Hafner Publishing Company.

Feo, T., & Resende, M. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, *6*, 109–133.

Fréville, A., & Plateau, G. (1994). An efficient preprocessing procedure for the multidimensional 0-1 knapsack problem. *Discrete Applied Mathematics*, *49*, 189–212.

Gavish, B., & Pirkul, H. (1982). Allocation of databases and processors in a distributed data processing. In J. Akola (Ed.), *Management of distributed data processing, North-Holland, Amsterdam*, 215–231.

Gavish, B., & Pirkul, H. (1985). Efficient algorithms for solving multiconstraint zero-one knapsack problems to optimality. *Mathematical Programming*, 31, 78–105.

Gilmore, P. C., & Gomory, R. E. (1966). The theory and computation of knapsack functions. *Operations Research*, *14*, 1045–1074.

Glover, F. (1965). A multiphase-dual algorithm for the zero-one integer programming problem. *Operation Research*, *13*, 879–919.

Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decision Science*, *8*, 156–166.

Glover, F. (1990). Tabu search. Part II. ORSA. *Journal of Computing*, *2*, 4–32.

Glover, F., & Kochenberger, G. (1996). Critical event tabu search for multidimentional knapsack problems. In I. H. Osman, & J. P. Kelly (Eds.), *Meta-heuristics: Theory and applications* (pp. 407–427). Dordecht: Kluwer Academics Publishers, 407–427.

Greenberg, H., & Pierskalla, W., (1970). Surrogate mathematical programs. *Operations Research*, 18, 924–939.

Hanafi, S., & Fréville, A. (1998). An efficient tabu search for the 0-1 multidimensional knapsack problem. *European Journal of Operational Research*, *106*, 659–675.

Hanafi, S., Fréville, A., & El Abdellaqui, A. (1996). Comparison of heursitics for the 0-1 multidimensional knapsack problem. In I. H. Osman, & J. P. Kelly (Eds.), *Meta-heuristics: Theory and applications* (pp. 449–465). Dordecht: Kluwer Academics Publishers, 449–465.

Haul, C., & Voss, S. (1997). Using surrogate constraints in genetic algorithms for solving multidimensional knapsack problems. In D. L. Woodruff, *Advances in computational and stochastic optimization, logic programming, and heuristic search. Interfaces in computer science and operation research* (pp. 235–251). Dordecht: Kluwer Academic Publishers, 235–251.

Hochbaum, D. S. (1996). *Approximation algorithms for NP-hard problems*. New York: PWS Publishing Company.

Khuri, S., Bäck, T., Heikötter, J. (1994). The zero/one multiple knapsack problem and genetic algorithms. In *Proceedings of the 1994 ACM symposium on applied computing (SAC'94)* (pp.188–193). ACM Press.

Kyparisis, G., Gupta, S., & Ip, C. (1996). Project selection with discount returns and multiple constraints. *European Journal of Operational Research*, *94*(1), 87–96.

Lee, J. S., & Guignard, M. (1988). An approximate algorithm for multidimensional sero-one knapsack problems-a parametric approach. *Management Science*, *34*(3), 402–410.

Lokketangen, A., & Glover, F. (1998). Solving zero-one mixed integer programming problems using tabu search. *European Journal of Operations Research*, *106*, 624–658.

Loulou, R., & Michaelides, E. (1979). New greedy-like heuristics for the multidimensional 0-1 knapsack problem. *Operations Research*, *27*(6), 1101–1114.

Magazine, M. J., & Oguz, O. (1984). A heuristic algorithm for the multidimensional zero-one knapsack problem. *European Journal of Operational Research*, *16*, 319–326.

Martello, S., & Toth, P. (1990). *Knapsack problems: Algorithms and computer implementations*. New York: Wiley.

Moraga, R. J. (2002). *Meta-RaPS: An effective solution approach for combinatorial problems*. PhD thesis. Orlando, FL: University of Central Florida.

Moraga, R. J., DePuy, G., & Whitehouse, G. (2003). Meta-RaPS approach for solving the 0-1 multidimensional knapsack problem. In Y. Dessouky (Ed.), *Proceedings of the 31st international conference on computers and industrial engineering, San Francisco, California*, 173–178.

Osorio, M. A., Glover, F., & Hammer, P. (2003). Cutting and surrogate constraint analysis for improved multidimensional knapsack solutions. *Annals of Operations Research*, *117*, 71–93.

Petersen, C. C. (1967). Computational experience with variants of the Balas algorithm applied to the selection of R&D projects. *Management Science*, *13*(9), 736–750.

Pirkul, H. (1987). A heuristic solution procedure for the multiconstraint zero-one knapsack problem. *Naval Research Logistics*, *34*, 161–172.

Senyu, S., & Toyoda, Y. (1968). An approach to linear programming with 0-1 variables. *Management Science*, *15*, 196–207.

Shih, W. (1979). A branch and bound method for the multiconstraint zero-one knapsack problem. *Journal of Operation Research Society*, *30*, 369–378.

Skorobohatyj, G. (2002). *MP-TESTDATA: Integer/mixed-integer programming problems.* http://elib.zib.de/pub/Packages/mp-testdata/ip/index.html.

Toyoda, Y. (1975). A simplified algorithm for obtaining approximate solutions to zero-one programming problems. *Management Science*, *21*, 1417–1427.

Volgenant, A., & Zoon, J. A. (1990). An improved heuristic for the multidimensional 0-1 knapsack problems. *Journal of the Operational Research Society*, *41*, 963–970.

Weingartner, H. M. (1967). *Mathematical programming and the analysis of capital budgeting problems*. Chicago: Markham Publishing.

Weingartner, H. M., & Ness, D. N. (1967). Methods for the solutions of the multidimensional 0/1 knapsack problem. *Operation Research*, *15*, 83–103.