MULTI-ITEM SCHEDULING PROBLEM WITH OVERTIME: SURROGATE RELAXATIONS AND HEURISTICS

Nejib Ben-Kheder
American Airlines Decision Technologies
P.O. Box 619616
Dallas-Fort Worth Airport, TX 75261-9616

and

Candace Arai Yano
Department of
Industrial & Operations Engineering
University of Michigan
Ann Arbor, MI 48109-2117

Technical Report 91-19

July 1990

Multi-Item Scheduling Problems with Overtime: Surrogate Relaxations and Heuristics

Nejib Ben-Kheder
American Airlines Decision Technologies
P.O. Box 619616
Dallas-Fort Worth Airport, TX 75261-9616
USA

Candace Arai Yano
Dept. of Industrial & Operations Engineering
University of Michigan
Ann Arbor, MI 48109-2117
USA

July 1990

Multi-Item Scheduling Problems with Overtime: Surrogate Relaxations and Heuristics

ABSTRACT

We address a multi-item, multi-period scheduling problem in which capacity can be increased by using overtime , which incurs an additional cost. The objective is to minimize the total cost of the schedules for the individual items, and overtime. We investigate a surrogate relaxation of the problem in which the multi-period problem is transformed into a single-period problem using surrogate multipliers. We show that for the two-constraint (two-period) problem, the multipliers can be computed exactly. The solution for the two-constraint problem is incorporated into an iterative heuristic for computing multipliers for problems with more than two constraints. The surrogate multipliers provide lower bounds, and serve as the basis for a good heuristic procedure. Computational results are reported.

Multi-Item Scheduling Problems with Overtime: Surrogate Relaxations and Heuristics

1. Introduction

We consider the problem of finding the best set of schedules, one for each of many items, to satisfy production requirements over multiple time periods. The objective is to minimize the total cost of the individual schedules and overtime. We formulate the problem as a multiple choice problem with side constraints which correspond to the resource constraint for each period. By allowing overtime, the capacity in each period becomes a decision variable and the problem is said to be a variable resource (VR) problem. The fixed resource (FR) version of the problem is a multidimensional 0-1 knapsack problem with multiple choice constraints.

Manne (1958) first formulated a multi-item scheduling problem as a multiple choice problem. He suggested solving the continuous version and showed that a near-optimal solution is obtained, provided the number of items far exceeds the number of periods. In such a case, the number of fractional variables is shown to be limited. Solutions to the original problem are obtained by rounding the solution of the linear programming (LP) relaxation, or by interpreting the fractional solution as a convex combination of candidate schedules. Bitran and Matsuo (1986) derived bounds on the error resulting from Manne's approximation schemes for both the fixed and variable resource problems. Multi-item lot-sizing problems usually involve scheduling the production of hundreds of items for which a large number of potential schedules can be generated. Dzielenski and Gomory (1965) suggested solving Manne's large LP by Dantzig-Wolfe decomposition. This method does not guarantee a principally integer solution until optimality. This difficulty is resolved in the Lasdon and Terjung (1971) revised simplex procedure. The principal disadvantage of the LP relaxation is its limitation to the case where the number of periods is negligible compared to the number of items, and the existence of split lots (fractional solution). It is not obvious how to modify such a solution to obtain good operational schedules.

We consider situations where the number of items is limited (e.g., less than one hundred) and a small number of alternative schedules are identified for each item. In these cases, the number of fractional variables in the LP solution may be relatively high. Since the problem is NP-complete and remains too large (potentially thousands of 0-1 variables) to be solved to optimality, we concentrate our efforts on developing simple heuristic procedures and a good lower bounding procedure to evaluate the heuristic solutions.

We suggest a lower bounding procedure based on a surrogate relaxation in which the multi-period problem is transformed into a single-period problem using surrogate multipliers. The single-constraint problem is a variant of the one-dimensional knapsack problem where a penalty cost is incurred for exceeding the knapsack capacity. If the optimal surrogate multipliers are known, then the bound generated by the surrogate relaxation is known to be better than or equal to the bounds generated by Lagrangian and linear relaxations (Glover 1975). We show that for a two-constraint (2C) problem the multipliers can be computed exactly. The solution for the 2C-problem is incorporated into an iterative heuristic for computing multipliers for problems with more than two constraints. We also develop several heuristics which use these surrogate multipliers.

In the remainder of this paper, we first state the multiple choice formulation of the problem and discuss the relevant literature on multidimensional knapsack problems. We then define the surrogate problem and suggest an approach to derive good surrogate multipliers. We will also discuss how to adapt existing solution procedures for the single knapsack, fixed-resource problem to solve the variable resource case, and how to derive good solutions for the multi-constraint problem from the analysis of the one-constraint problem. Finally, we test the effectiveness of our lower bounding scheme and the quality of the heuristic solutions.

2. The multi-item scheduling problem (MISP): a multiple choice formulation

The MISP problem can be stated as:

$$\text{Minimize } \sum_{i=1}^{S} \sum_{k=1}^{K_i} C_{ik} Y_{ik} + \sum_{h=1}^{H} W_h V_h \tag{1}$$

$$\text{s.t.} \quad \sum_{i=1}^{S} \sum_{k=1}^{K_i} L_{ikh} Y_{ik} \le R_h + V_h, \qquad h=1,..,H; \tag{2}$$

$$\sum_{k=1}^{K_i} Y_{ik} = 1, \qquad i=1,..,S; \tag{3}$$

$$Y_{ik} = 0 \text{ or } 1, \qquad i=1,..,S;\ k=1,..,K_i; \tag{4}$$

$$V_h \ge 0, \qquad h=1,..,H; \tag{5}$$

where

S: total number of items,

K_i: total number of candidate schedules for item i,

Y_{ik}: 0-1 variable equal to 1 if schedule k is assigned to item i and 0 otherwise,

L_{ikh}: workload induced by item i-schedule k in period h,

C_{ik}: cost of item i-schedule k,

V_h: overtime in period h

R_h: regular time capacity in period h, and

W_h: cost per unit of overtime in period h.

Constraints (2) are the variable resource constraints; they ensure that overtime is incurred whenever the total workload exceeds the regular time capacity for a given period. Constraints (3) are the multiple choice constraints, which in combination with constraints (4), state that one and only one schedule is chosen for each item. Constraints (5) imply that no cost savings or penalties are incurred when undertime occurs in a period.

Most of the research on multiconstraint knapsack problems deals with the 0-1 problem without multiple choice constraints. Early work by Gilmore and Gomory (1966) and Weingartner and Ness (1967) focuses on dynamic programming (DP) based algorithms. More recently, Shih (1979) and Gavish and Pirkul (1985) develop branch and bound (B&B) algorithms that take advantage of the structure of the problem, and are easier to conceptualize and more convenient to use than previous algorithms. The efficiency of the B&B procedures depends strongly on the lower bounding scheme. Gavish and Pirkul (1985) present several methods for computing lower bounds, including Lagrangian, surrogate and composite relaxation. In extensive computational results, it is shown that for the 0-1 integer problem with few knapsack constraints, surrogate relaxation is a viable alternative to the more commonly used Lagrangian and LP relaxations.

Once a set of multipliers is computed, we need to solve a one-dimensional (single-constraint) multiple choice knapsack problem (SMCK). Due to the large number of applications of this problem (e.g., menu planning, capital budgeting and catalog planning), extensive theoretical and empirical investigations have led to the development of highly efficient solution procedures. DP procedures have been suggested by Dudzinski (1984) and Bean (1987). Alternatively, Nauss (1978) suggest a B&B procedure based on a relaxation of the multiple choice constraints. Sinha and Zoltners (1979) and Dyer et al. (1984) use the LP relaxation, for which linear time algorithms have been proposed by Zemel (1984), Dyer (1984) and Dudzinski and Walukiewicz (1984).

In the next section, we extend some of the theoretical results of Gavish and Pirkul to our problem and suggest various heuristics to compute surrogate multipliers. We also discuss how to adapt the fixed-resource SMCK algorithm to solve the variable-resource case.

3. Surrogate relaxation of MISP

In the following, we will use a matrix representation of the problem. We rewrite MISP as:

$$\begin{array}{ll} \text{Min} & CY + WV \\ \text{s.t.} & LY \le R + V \\ & Y \in \Psi,\ V \in R^{+} \end{array} \qquad \text{(P)}$$

where

Y is the decision vector with K 0-1 elements,

V is the overtime vector with H non-negative elements,

L is the H x K workload matrix,

C and R are the cost and dock capacity vectors of conforming dimensions, respectively, and

Ψ is the set defined by all 0-1 and multiple choice constraints in the decision vector Y.

We will also denote by L_h the workload vector in period h.

The surrogate relaxation of problem P associated with a given set of multipliers υ (≥ 0) is

$$\begin{array}{ll} \text{Min} & CY + WV \\ \text{s.t.} & \upsilon(LY - R - V) \le 0 \\ & Y \in \Psi,\ V \in R^{+} \end{array} \qquad (SP_\upsilon)$$

Let z($\cdot$) be the value of an optimal solution to problem ($\cdot$). It is easy to show that $z(SP_\upsilon)$ is a lower bound on z(P). The best bound is given by the solution to the surrogate dual:

$$\text{Sup}_{\ \upsilon \ge 0}\ z(SP_\upsilon). \qquad \text{(S)}$$

The idea of surrogate relaxation was first introduced by Glover (1965) but did not receive as much attention as the more widely known Lagrangian relaxation, for which very simple search procedures have been developed and extensively tested on a variety of optimization problems. Considerably less experience has been accumulated with surrogate relaxation. Renewed interest in surrogate relaxation has been triggered by the work of Karwan and Rardin (1979), who show that the surrogate bound is, in general, strictly tighter than the Lagrangian bound. In a more recent paper (1984), they adapt some of the ideas in a Lagrangian dual search (e.g., subgradient optimization) and develop several general surrogate multiplier search procedures. We propose an alternate search method based on an efficient procedure to derive the optimal surrogate multipliers for the two-constraint problem. Our procedure is similar to the one developed by Gavish and Pirkul (1985) for the 0-1 problem, which has also been applied to the general knapsack problem by Pirkul and Narasimhan (1986).

3.1 Derivation of optimal multipliers for a two-constraint problem

We consider the following two-dimensional knapsack problem:

$$\text{Min } CY + W_1 V_1 + W_2 V_2 \tag{P2}$$

$$\text{s.t.} \quad L_1 Y \le R_1 + V_1 \tag{6}$$

$$L_2 Y \le R_2 + V_2 \tag{7}$$

$$Y \in \Psi$$

$$V_1, V_2 \ge 0$$

and its surrogate relaxation for a pair of multipliers (γ, μ):

$$\text{Min } CY + W_1 V_1 + W_2 V_2 \tag{SP$\gamma\mu$}$$

$$\text{s.t.} \quad (\gamma L_1 + \mu L_2)\, Y \le (\gamma R_1 + \mu R_2) + (\gamma V_1 + \mu V_2)$$

$$Y \in \Psi$$

$$V_1, V_2 \ge 0$$

We assume that constraints (6) and (7) are linearly independent. We also assume that constraint (6) is violated by the solution to SP_{01}, and that constraint (7) is violated by the solution to SP_{10} , i.e., both γ^* and μ^* are positive. Otherwise an optimal solution to the surrogate problem has been found. The dual problem can be rewritten as:

$$\text{Sup}_{\mu>0}\, z(SP_{1\mu}) \tag{D2}$$

In the following we investigate some of the properties of the solution to $SP_{1\mu}$. Proofs of the lemmas appear in the Appendix.

<u>Lemma 1</u>: Let $\mu < W_2/W_1$. Then an optimal solution to $SP_{1\mu}$ has $V_2=0$. Similarly, if $\mu > W_2/W_1$ then an optimal solution to $SP_{1\mu}$ has $V_1=0$.

Suppose $\mu < W_2/W_1$. As a result of Lemma 1, $SP_{1\mu}$ can be rewritten as:

$$\text{Min } C\,Y + \mathbf{w}\,\mathbf{v}$$

$$\text{s.t.} \quad \mathbf{a}\,Y \le \mathbf{r} + \mathbf{v}$$

$$Y \in \Psi,\ V \ge 0$$

where $w=W_1$, $v=V_1$, $r=R_1+\mu R_2$ and $a=L_1+\mu L_2$. In the remainder of this paper, we will refer to this problem as the surrogate multiple choice knapsack problem, SMCK(a,r,w).

In the following, for simplicity, we will assume that $\mu < W_2/W_1$. Similar results hold for the opposite case ($\mu>W_2/W_1$), and can be obtained by reindexing the constraints.

Lemma 2: Let (Y,V) be an optimal solution to $SP_1\mu$. Then at most one constraint is violated by this solution.

Lemma 3: Let (Y,V) be an optimal solution to $SP_1\mu$ that does not satisfy constraint (1). Then $\mu^* \leq \mu$, where μ^* denotes the optimal multiplier.

Lemma 4: Let (Y,V) be an optimal solution to $SP_{1\mu}$ that does not satisfy constraint (2). Then $\mu^* \geq \mu$.

Lemmas 2, 3 and 4 identify an improving direction for the dual problem when one of the constraints is violated.

For the special case where $\mu=W_2/W_1$, $SP_{1\mu}$ has an infinite number of solutions in terms of overtime values (V_1 and V_2). Let (Y^*,V_1^*,V_2^*) be one such solution and let $v=V_1^*+\mu V_2^*$. Then any feasible solution (Y^*,V_1,V_2) that satisfies $v=V_1+\mu V_2$ is also optimal. To check the feasibility of the surrogate solution with respect to the original problem P2, we use the following overtime values: $V_1=\mathrm{Min}\{v, \mathrm{Max}\{0, L_1Y^*-R_1\}\}$ and $V_2=(v-V_1)/\mu$. If $L_1Y^* \leq R_1 + V_1$ and $L_2Y^* \leq R_2 + V_2$ then $\mu^*=\mu$. Otherwise it can be shown that at most one of the constraints is violated and a search direction is then identified using results similar to Lemmas 3 and 4.

The theoretical results established in the preceding lemmas lead to the following procedure to compute an ε-interval that contains the optimal multipliers.

Procedure SI: (single multiplier, integer knapsacks)

Step 0:

(a) Solve SMCK(L_2,R_2,W_2).
If the solution satisfies $L_1Y \leq R_1$ then set $\gamma^*=0$ and $\mu^*=1$, and terminate.
Otherwise let $\gamma^*=1$ and go to (b).

(b) Solve SMCK(L_1,R_1,W_1).
If the solution satisfies $L_2Y \leq R_2$ then set $\mu^*=0$ and terminate.
Otherwise go to Step 1.

Step 1:

Let $\mu = W_2/W_1$. Solve SMCK($L_1+\mu L_2$, $R_1+\mu R_2$,W_1) .
Let $V_1 = \mathrm{Min}\ \{v, \mathrm{Max}\{0, L_1Y-R_1\}\}$ and let $V_2 = (v-V_1)/\mu$.
If $L_1Y \leq R_1 + V_1$ and $L_2Y \leq R_2 + V_2$ then set $\mu^*=W_2/W_1$ and terminate.
If $L_1Y > R_1 + V_1$ then set $\mu_{max}=W_2/W_1$ and $\mu_{min}=0$, and go to Step 2.
If $L_2Y > R_2 + V_2$ then set μ_{max}=M (large value) and $\mu_{min}=W_2/W_1$, and go to Step 2.

Step 2:

If $(\mu_{max} - \mu_{min}) \leq \varepsilon$, terminate.

Otherwise go to Step 3.

Step 3:

Update μ: $\mu = \mu_{min} + (\mu_{max} - \mu_{min})/2$.

Solve SMCK($L_1+\mu L_2$, $R_1+\mu R_2$,w), where $w=\text{Min}\{W_1, W_2/\mu\}$.

If $w=W_1$ then let $V_1=v$ and $V_2=0$.

Otherwise let $V_1=0$ and $V_2=v/\mu$.

If $L_1Y \leq R_1+V_1$ and $L_2 \leq R_2 + V_2$ then set $\mu^*=\mu$ and terminate.

If $L_1Y > R_1 + V_1$ then set $\mu_{max}=\mu$ and go to Step 2.

If $L_2Y > R_2 + V_2$ then set $\mu_{min}=\mu$ and go to Step 2.

The performance of this iterative procedure depends on how μ is updated at each iteration. Here, we have suggested a bisection method. If the initial interval has a range $\Delta\mu$ and if the tolerance is equal to ε, then the algorithm would require $O(\log_2(\Delta\mu/\varepsilon))$ iterations before termination. Alternatively, if a golden section search is performed then the interval is reduced by the ratio $(2/(1+\sqrt{5}))$ at each iteration.

At each iteration of SI, a one-dimensional multiple choice knapsack problem would have to be solved to optimality, which can be time consuming. It may be preferable to compromise on the quality of the dual bounds for the sake of computational efficiency. We suggest solving the linear relaxation of SMCK, LSMCK, at each iteration. The multipliers obtained by this procedure can be shown to equal the dual multipliers of the LP relaxation of the original two-constraint problem. Therefore the bound is equal to the LP bound. A better bound can be obtained by using the dual multipliers to generate a surrogate constraint and solving the resulting surrogate problem as a 0-1 problem. In the following, we will refer to the modified SI procedure, where SMCK is replaced by LSMCK, as SL (single multiplier, linear knapsacks). Besides being solvable in a linear time, the continuous knapsack problem, LSMCK, offers the possibility of using sensitivity analysis and an efficient reoptimization procedure to solve successive problems.

The LSMCK problem: solution approach

As mentioned earlier the LSMCK problem with fixed resources has received a great deal of attention (see Dudzienski and Walukiewicz (1987) for a recent literature review). The problem is usually solved in two phases. First, some of the variables are eliminated from the problem (set equal to zero or one) using dominance criteria. This reduction consists of sorting the variables in the same multiple choice set according to their cost-to-weight ratios. This operation, which constitutes the time consuming part of the reduction scheme, can be performed in running time $O(\sum_{i=1}^{S} K_i \log K_i)$. In the next step, the dual of the reduced problem is solved. It is shown that the optimal dual multiplier can assume only a finite number of values, which can be determined *a priori*. The set of potential dual multipliers is ordered and a median search for the optimal value is performed. At each iteration, the median is checked for optimality. If it is not optimal then it is shown that half of the values in the set (and therefore half the variables) can be eliminated. As a result, the search for the optimal value is reported (see Dyer 1984), for example) to be O(K'), where K' is the total number of variables in the reduced problem.

The dual of LSMCK may be formulated as:

$$\text{Min } r\delta - \sum_{i=1}^{S} \upsilon_i \qquad \text{(DSMCK)}$$

$$\text{s.t.} \quad \upsilon_i - a_{ik}\delta \le C_{ik}$$

$$0 \le \delta \le w$$

which can be rewritten as a minimization of a convex, piecewise linear function:

$$\begin{aligned}\text{Min } f(\delta) &= \text{Min}_{0\le\delta\le w}\,(r\delta - \sum_{i=1}^{S} \text{Min } \{C_{ik} + a_{ik}\,\delta,\ k=1,..,K_i\}\,) \\ &= \text{Min}_{0\le\delta\le w}\,(r\delta - \sum_{i=1}^{S} f_i(\delta)\,)\end{aligned}$$

where $f_i(\delta) = \text{Min } \{C_{ik} + a_{ik}\,\delta,\ k=1,..,K_i\}$

It is easy to see from the structure of $f(\delta)$ that a minimum occurs at either a breakpoint $\delta = \frac{C_{ik} - C_{ik'}}{a_{ik} - a_{ik'}}$ $(0 < \delta < w)$ for some multiple choice set i, or at one of the bounds (0 or w). The two bounds correspond to trivial solutions which can be checked easily for optimality.

<u>Trivial solutions</u>:

1. Let $f_i = \text{Min}_{1\le k\le K_i} C_{ik}$ and let $k'(i) = \text{argmin } \{ a_{ik},\ 1\le k\le K_i \mid C_{ik}=f_i \}$. The minimum cost alternative in set i is k'(i). Ties are broken by choosing the alternative with minimum total resource consumption. This solution is optimal if it results in no overtime, i.e., if $\sum_{i=1}^{S} a_{ik'(i)} \le r$, then an optimal solution to LSMCK is given by

$Y_{ik'(i)} = 1, \quad i=1,..,S$

$Y_{ik} = 0, \quad$ otherwise

and $v = 0$.

2. Let $g_i = \text{Min } \{C_{ik} + w\, a_{ik},\ 1 \le k \le K_i\}$ and $k''(i) = \text{argmax } \{ a_{ik},\ 1 \le k \le K_i \mid C_{ik} + wa_{ik} = g_i \}$. Assuming that the solution results in overtime, k"(i) is the minimum (total) cost alternative in set i. The ties are broken by choosing the alternative with minimum C_{ik}. This solution is optimal if it results in positive overtime, i.e., if $\sum_{i=1}^{S} a_{ik''(i)} > r$, then an optimal solution to LSMCK is given by

$Y_{ik''(i)} = 1 \quad i=1,..,S,$

$Y_{ik} = 0$ otherwise,

and $v = r - \sum_{i=1}^{S} a_{ik''(i)}$.

This is the only case where the optimal solution has positive overtime.

Both trivial solutions are integer, and are therefore optimal to the integer problem as well. In general, the solution to the continuous problem has two fractional variables. Next we describe how the solution is determined in non-trivial cases.

Optimality conditions:

The piecewise-linear convex function, $f(\delta)$, achieves its minimum at δ^* if its left gradient, $\partial f^-(\delta^*)$, is non-positive and its right gradient, $\partial f^+(\delta^*)$, is non-negative. The gradients at δ^* are:

$$\partial f^-(\delta^*) = r - \sum_{i=1}^{S} \text{Max } \{a_{ik} \mid C_{ik} + \delta^* a_{ik} = f_i(\delta^*) \}$$

and

$$\partial f^+(\delta^*) = r - \sum_{i=1}^{S} \text{Min } \{a_{ik} \mid C_{ik} + \delta^* a_{ik} = f_i(\delta^*) \} .$$

The minimum is obtained at a breakpoint $\delta^* = \dfrac{C_{i^*k''} - C_{i^*k'}}{a_{i^*k'} - a_{i^*k''}}$ for some set i*, and some variable indices k' and k" in the set i* (we assume $a_{i^*k'} < a_{i^*k''}$).

The optimal solution to the primal problem is obtained as follows:

$$Y_{ik} = \begin{cases} 1 & \text{for } \{ k \mid f_i(\delta^*) = C_{ik} + \delta^* a_{ik} \text{ for } i \ne i^* \} \\ \alpha & \text{for } k=k'' \text{ and } i=i^* \\ 1-\alpha & \text{for } k=k' \text{ and } i=i^* \\ 0 & \text{otherwise,} \end{cases}$$

where $\alpha = \dfrac{r - \sum_{i \ne i^*} a_{ik} Y_{ik} - a_{i^*k'}}{a_{i^*k''} - a_{i^*k'}}$ is determined so that $\sum_{i \ne i^*} a_{ik} Y_{ik} + \alpha\, a_{i^*k''} + (1-\alpha)\, a_{i^*k'} = r.$

Dominance Criteria:

In the fixed resource case, it has been shown that the problem is reduced as follows.

For a given set i:

1. If $a_{ik} \leq a_{il}$ and $C_{ik} \leq C_{il}$, then there exists an optimal solution with $Y_{il} = 0$. In this case, any solution containing alternative k will cost less than any solution containing alternative l, since alternative l consumes more resources and costs more than alternative k.

2. If $a_{ik} < a_{il} < a_{ip}$ and $\frac{C_{ik} - C_{il}}{a_{il} - a_{ik}} < \frac{C_{il} - C_{ip}}{a_{ip} - a_{il}}$, then there exists an optimal solution with $Y_{il} = 0$. In this case, it can be shown that for any value, δ, of the dual multiplier, the reduced cost of alternative l, $C_{il} + \delta a_{il}$, is greater than either the reduced cost of alternative k or the reduced cost of alternative p. Therefore it is never selected.

Further variable elimination can be achieved in the variable resource version of the problem, where we can show that:

3. If $a_{ik} > a_{il}$ and $\frac{C_{il} - C_{ik}}{a_{ik} - a_{il}} \geq w$ then there exists an optimal solution with $Y_{il} = 0$. In this case, it can be easily shown that the reduced cost of alternative k, for any value of the multiplier δ ($\leq w$), is always less than the reduced cost of alternative l. Therefore alternative l is dominated by alternative k.

We conclude that the VR version of the problem can be solved by any existing dual algorithm developed for the FR case, with the addition of two features, namely, the second trivial solution and the third dominance criterion discussed above. As a result solution times for the VR problem are expected to be even smaller than those observed for the FR case.

Finding good multipliers for the two-constraint problem by procedure SL may require iteratively solving a large number of single-constraint problems. Although there are fast algorithms for these problems, it may be computationally tedious to solve the successive LPs to optimality without taking advantage of previous solutions. As in many dual-based integer programming approaches, the optimal solution to one surrogate problem remains optimal for a range of multiplier values, which can be determined by sensitivity analysis (see Ben-Kheder and Yano 1990 for details). The main idea is that the multipliers can be adjusted, thereby increasing the dual objective value, without a change in the optimal dual solution.

The bisection search in procedure SI starts with an initial interval large enough to avoid eliminating the optimal solution. With no prior knowledge of μ^*, and therefore no estimate of an initial upper bound on μ, the number of iterations may be quite large. Sensitivity analysis is used to initialize the search interval at some predetermined range ($=K\varepsilon$), and is also used to speed up the convergence to the optimal multiplier when the interval range is small (e.g., less than $k\varepsilon$.) The sensitivity analysis consists of computing a set of potential increments to the multiplier value and choosing the minimum one. The number of increments is proportional to the total number of alternatives.

The number of LPs explicitly solved within the solution procedure for LSMCK is at most $\log_2 K/k$, and therefore its value can be selected *a priori*. The choice of K and k depends on the magnitude of the increments generated by the sensitivity analysis. Generally, the size of the increment, i.e., the range of multipliers for which the present solution is optimal, tends to be high when the multiplier value is set at its lower bound (zero) or at its upper bound (infinity), but drastically declines as the search interval gets smaller. We need to choose the parameters K and k so that the time savings due to the reduction of the number of LPs solved is not offset by the time required to compute the increments. Procedure SL with the added sensitivity analysis is referred to as procedure SLS.

3.2. Derivation of multipliers for the multiconstraint problem

Define $SP^{(h)}_\gamma$ as follows:

$$\text{Min } CY + WV \qquad (SP^{(h)}_\gamma)$$

$$\text{s.t.} \quad \gamma LY \le \gamma R + \gamma V \qquad (M1)$$

$$L_h Y \le R_h + V_h \qquad (M2)$$

$$Y \in \Psi,\ V \in R^+$$

where constraint M1 is the single constraint of the surrogate problem SP_γ for a set of multipliers γ, and constraint M2 is the h^{th} period constraint of the original problem P. $SP^{(h)}_\gamma$ is a two-constraint problem to which the analysis of section 3.1 can be applied. Consider the surrogate relaxation of $SP^{(h)}_\gamma$ where the multiplier of the first constraint is equal to 1 and the multiplier of the second constraint is equal to μ. Lemma 4 states that if the solution to the surrogate problem violates the second constraint then a better surrogate bound can be obtained by increasing μ. Therefore if the solution to SP_γ violates some constraint, h, of the multi-constraint problem P, then an ascent direction is identified for the surrogate dual. A better surrogate bound is obtained by increasing the h^{th} component of γ.

Based on this result, we incorporate the two-constraint procedure SL in a heuristic to derive surrogate multipliers for the multi-constraint problem. The following procedure is similar to the one suggested by Gavish and Pirkul for the 0-1 multidimensional knapsack problem.

Procedure ML: (multiple multipliers, linear knapsacks)

Step 0. Determine the constraint that produces the highest objective value when all other constraints are ignored in problem P. Reindex this constraint as constraint 1.

Step 1. Determine the constraint most violated by the solution of the single-constraint problem using constraint 1. Reindex this constraint as constraint 2.

Step 2. Apply procedure SL (or SLS) to the problem determined by constraints 1 and 2. If the bound does not improve, terminate. Otherwise let the surrogate constraint be constraint 1 and go to step 1.

In order to implement procedure ML, we need to resolve some details. For example, it is not obvious how to define the appropriate (aggregate) overtime costs for the two-constraint problem, nor is it clear what "most violated" means in the presence of available overtime.

1. Transforming $SP^{(h)}{}_{\gamma}$ into a standard two-constraint problem:

Ambiguities exist due to the presence of V_h in both constraints M1 and M2. It is not clear how to define the overtime costs corresponding to the first and second constraints in $SP^{(h)}{}_{\gamma}$, or how to interpret the aggregate overtime in problem $SP^{(h)}{}_{\gamma}$ in terms of the overtime in periods 1,..,H.

Let us first consider the surrogate problem corresponding to $SP^{(h)}{}_{\gamma}$ for some multiplier μ. This problem is actually $SP_{\gamma'}$, where $\gamma'_h=\gamma_h+\mu$ and $\gamma'_j=\gamma_j$ otherwise. It is easy to show that for the one-dimensional problem $SP_{\gamma'}$, $w = W_{j^*} = \mathrm{Min}_{1\leq j\leq H}\ W_j/\gamma'_j$. Moreover if v is the aggregate overtime solution in problem $SP_{\gamma'}$, then in terms of the original overtime variables, we have $V_{j^*}=v/\gamma'_{j^*}$ and $V_j=0$ otherwise. To check the feasibility of the solution to $SP\gamma'$ in the two-constraint problem, $SP^{(h)}{}_{\gamma}$, and thereby identify an improving direction, we need to define the overtime for constraints M1 and M2 in terms of v. Based on the results above, we have:

$V_1 = \gamma_{j^*}V_{j^*} = \gamma_{j^*}/v\gamma'_{j^*}$ for constraint M1

and $V_2 = V_h$ for constraint M2.

The feasibility check consists of verifying that:

$(\gamma L + \mu L_h)\ Y \leq (\gamma R + \mu\ R_h) + V_1$ for constraint M1

and $L_h R \leq R_h + V_2$ for constraint M2.

Notice that unlike the standard two-constraint problem, both the first and second constraint may have positive overtime (if $j^*=h$). For this reason, modifications of the two-constraint sensitivity analysis procedure are required, but they are not described in this paper for the sake of brevity.

2. Determination of the most violated constraint:

Since overtime is allowed and not bounded, feasibility is not an issue in problem P. The "constraint violation" is defined in terms of the overtime cost. The most violated constraint, h^*, is defined as the constraint h with the highest overtime cost resulting from the solution under consideration, i.e., $h^* = \text{argmax } \{W_h V_h,\ h=1,..,H\}$.

3. Avoiding redundant computations in ML:

The initialization step in SL need not be performed when SL is used in conjunction with procedure ML. At each iteration in ML, we solve the surrogate dual of $SP^{(h)}{}_{\gamma}$, where γ is the set of multipliers determined in the previous iteration. In SL, when we first let the multiplier corresponding to the first constraint be equal to zero, the resulting problem has already been solved in step 0 of ML, where all constraints are ignored except one. When we let the multiplier corresponding to the second constraint be equal to 0, the problem obtained is SP_{γ}, which has been solved in the previous iteration.

At termination, the bound obtained by successively solving the two-constraint problems will be at best equal to the LP bound. A better bound is obtained by solving the surrogate dual problem as an integer problem. The dual multipliers determined by procedure ML are used to generate a surrogate constraint, and the single-constraint multiple choice problem obtained is then solved as a 0-1 problem by a specialized branch and bound procedure.

Computational experience shows that the unidirectional search procedure ML provides good multiplier values in a reasonable amount of time. The quality of the multipliers is tested both in terms of tightness of the bounds and in terms of their ability to reflect the resource constraints in an aggregate manner. This last property can be judged by the quality of the heuristic solutions that make use of the surrogate multipliers to solve the multidimensional problem as a one-dimensional problem.

4. Heuristic Solutions

A solution to the multidimensional problem can be obtained, with no additional computation, from the solution to the one-dimensional surrogate problem. If it is solved as a 0-1 problem then the solution is integer and no rounding is needed. Feasibility is not an issue here, since overtime is allowed, but the actual overtime values for each period are needed to evaluate the actual cost of the solution. This solution will be referred to as the single-constraint integer knapsack heuristic (SIKH), which requires solving a 0-1 problem to optimality using the multipliers from procedure ML. We suggest an alternate heuristic procedure which is executed in parallel with procedure ML. Each time a continuous single-knapsack problem is solved in Step 2 of procedure ML, we use its fractional solution to construct a 0-1 solution to the original problem. Let γ be the set of current multipliers, and let b be the matrix of current surrogate coefficients ($b=\gamma A$). Let Y^* be the solution to the current surrogate problem. From the results in Section 3, we know that there are at most two fractional values and they belong to the same multiple-choice set. Let m be the index of this set. (If such a set does not exist then the solution is integer and no rounding is needed.) Also let $Y_{mj'}$ and $Y_{mj''}$ be the two fractional values (where $b_{mj'} < b_{mj''}$). Procedure MLKH (multiple linear knapsack heuristic) uses this information to generate a set of 0-1 solutions and chooses the least cost one.

<u>Procedure MLKH</u>:

Step 0. Let γ be the set of surrogate multipliers at step 2 of procedure ML. Let b be the corresponding matrix of surrogate coefficients and let Y^* be the optimal solution to the current (continuous) surrogate problem. Let m, j'and j'' be as defined above.

Step 1. Generation of an initial 0-1 solution:

For all k such that $b_{mj'} \le b_{mk} \le b_{mj''}$, construct the schedule Y(k) as follows:

Let $Y_{ij}(k) = Y_{ij}^*$ for $i \neq m$, $j=1,..,K_i$,

$Y_{mj}(k) = 0$ for $j \neq k$, and

$Y_{mk}(k) = 1$.

Find $k^* = k$ such that the total cost of schedule Y'(k) is minimum.

Let $Y'=Y(k^*)$.

Step 2. Reduction of overtime:

Find the period h^* with the maximum overtime cost if schedule Y' is implemented,
(i.e., $h^* = \text{argmax}\{ W_h \text{Max} \{ L_h Y' - R_h, 0\} , h=1,..,H \}$.)

Find the item i^*, with the largest workload in the period h^* if Y' is implemented,
(i.e., $i^* = \text{argmax}\{ \text{Max}\{L_{ijh^*}, j=1,..,K_i \}, i=1,..,S \}$).

For all alternatives in the multiple choice set corresponding to i^*, construct the schedule Y(k) as follows:

Let $Y_{ij}(k) = Y_{ij}'$ for $i \neq m$, $j=1,..,K_i$,

$Y_{i^*j}(k) = 0$ for $j \neq k$, and

$Y_{i^*k}(k) = 1$.

Find k* such that Y(k) is the least cost schedule. Let Y" = Y(k*). Go to step 3.

Step 3. If the cost of schedule Y" is lower than that of the incumbent solution then update the incumbent to be Y". Terminate.

Remark: The incumbent solution is initialized at the trivial solution which consists of choosing the least cost alternative (based on the C_{ik}s) in each multiple choice set. This solution is referred to as the ARGMIN solution.

Conceptually, MLKH is an enumeration procedure. An alternative is selected *a priori* for each multiple choice set, except in one "critical set" for which a set of alternatives is evaluated. In step 1 the critical set is defined as the unique set with fractional variables in the solution to the surrogate problem. We limit our search to alternatives with an aggregate load between the aggregate loads of the fractional variables. We have observed that the solution to the 0-1 version of the surrogate problem usually has this property. The selected alternative then has a smaller cost than the lower workload alternative (of the two fractional variables) and a smaller load than the lower cost alternative. The solution generated in step 1 is further improved by the overtime reduction scheme in step 2. First, we identify a "critical period" and define the "critical set" as the one with the largest load in the critical period if the policy determined in step 1 is implemented. We consider all the possible alternatives corresponding to this set while using the solution from step 1 for the remaining sets. We choose the schedule with the least total cost.

The search for the critical period and the critical set consists of comparing a number of values at most equal to the total number of alternatives plus the number of periods. The total number of alternatives investigated in Steps 1 and 2 does not exceed the number of alternatives in the critical set. Therefore the additional amount of computation needed to generate the heuristic solution is linear in the problem size.

Along with these heuristics that make use of the multipliers derived by procedure ML, we mention two simpler heuristics which do not require the explicit computation of surrogate multipliers: (i) the ARGMIN heuristic introduced above, which does not take into account the overtime costs, and (ii) the most violated constraint heuristic (MVH) which has the same features as MLKH with the distinction that it is only run once for a specific set of surrogate multipliers. The surrogate constraint is the constraint for the period with the highest overtime cost in the ARGMIN solution, i.e., the surrogate constraint is given by $L_{h^*}Y \leq R_{h^*} + V_{h^*}$, where $h^*=\text{argmax}\{ W_h \text{Max} \{ L_hY'-R_h,0\} , h=1,..,H \}$ and Y' is the ARGMIN solution.

In addition to producing lower and upper bounds for the multidimensional problem, good surrogate multipliers can be used to reduce the problem size, as we explain in the next section.

5. Computational experience

In order to test the heuristics and the effectiveness of the surrogate lower bounding scheme, we run a series of experiments. All CPU times are given for a workstation with an 80386 processor running at 16 MHZ. The computer codes are written in Turbo Pascal version 5.0.

All the procedures developed in this paper can be applied to any multi-constraint multiple-choice problem provided all the constraints have non-negative coefficients and a penalty cost is defined for violating each constraint. It is clear from the terminology adopted so far that we are primarily interested in scheduling problems. The problems generated in our computational experiments have the structure of scheduling problems. For a given item, all the alternative schedules have the same total workload over the planning horizon, i.e., the same total production time is required over the planning horizon. The schedules differ in their individual costs and the distribution of the workload over the planning horizon. Some schedules will have items produced in large batches so as to benefit from economies of scale in production. Other schedules will have just-in-time type schedules where the inventory costs are minimized. Next, we discuss how to generate the cost matrix and the workload matrix.

Experimental set:

The costs of the individual schedules are generated from the uniform distribution U[50,150]. The workload matrix for a given item i, i=1,..,S, is obtained by the following scheme. First the total workload, T_i is generated from the uniform distribution U[10H,100H]., Then for each schedule j, j=1,..,K, we generate a vector of coefficients α_{ijh}, h=1,..,H, from U[1,50]. The workload of alternative j for item i in period h is given by $\frac{\alpha_{ijh} T_i}{\sum_h \alpha_{ijh}}$. This ensures that the ratio of the maximum workload to the minimum workload for a given alternative is between one (balanced schedule) and fifty (lumpy schedule).

The regular time capacity for each period is computed as the average of the maximum and the minimum possible workload in that period. Finally the overtime cost parameter is computed by first specifying the value of β, the proportion of total cost represented by overtime cost. Consider the ARGMIN solution. If this set of schedules results in no overtime, then it is optimal (in which case all the alternatives forming the ARGMIN schedule are deleted in order to avoid this trivial solution). Otherwise let V be the total overtime computed over all periods. A generic overtime cost, W, is then given by the equality: $W\ V = \beta(W\ V + \sum_{i=1}^{S} C_{ik^*})$, where $k^* = \text{argmin}\{ C_{ik}, k=1,..,K\}$ for a given set i, i=1,..,S. The unit overtime cost for each period is then given by U[0.5W,2W].

We report both the quality of the solutions and computation times. First, the heuristics are compared for various data sets. We analyze the effect of the problem size and the overtime cost on the relative performance of each heuristic. We also investigate the quality of the lower bound obtained by solving the surrogate problem using the set of multipliers from procedure ML. The gap between the lower and upper bounds provided by the best heuristic solution is reported as a measure of the effectiveness of the ML procedure.

Comparison of the heuristics:

Two types of heuristics were introduced in section 4 . The first group relies on the derivation of a set of multipliers by procedure ML, and includes SIKH and MLKH. We also suggested two simpler heuristics, ARGMIN and MVH, that can be run independently of procedure ML.

We conducted a computational experiment consisting of 180 problems. For each set of problem parameters, ten problems are generated by varying the schedule costs, total workloads and degree of lumpiness. For each heuristic, the average relative deviation from the MLKH solution, which was the best heuristic in all but one case, is reported.

In the first experiment we study the effect of the problem size (number of alternatives, K, and number of multiple choice sets, S). The overtime factor, β, is set to 0.5 and the number of side constraints is set to five . The results are shown in Table 1.

Problem Size		Heuristics		
S	K	ARGMIN	MVH	SIKH
25	10	28.4	11.5	0.8
	20	56.3	15.9	1.6
	30	30.9	13.0	1.9
	40	39.2	16.7	2.6
10	25	31.3	11.9	3.2
20		19.6	8.6	1.6
30		31.5	14.0	1.3
40		36.9	20.3	2.6

Table 1: Effect of the problem size on the relative performance of the heuristics (% deviation from the MLKH heuristic solution.)

The second experiment deals with the effect of the number of side constraints. The problems have 40 multiple choice sets with 20 alternatives in each set. The overtime cost factor is set to 0.5. The results are reported in Table 2.

	Heuristics		
# of side constraints (H)	ARGMIN	MVH	SIKH
3	48.6	14.0	1.4
4	24.7	12.0	2.2
5	23.1	9.7	0.5
6	29.5	17.7	3.4
8	28.9	15.9	1.5
10	26.5	16.8	1.0

Table 2: Effect of the number of knapsack constraints on the relative performance of the heuristics (% deviation from the MLKH heuristic solution.)

Finally, the heuristics are compared for different values of the overtime factor. The problems have 45 sets with 25 alternatives each, and 5 side constraints. The results are summarized in Table 3.

	Heuristics		
Overtime cost factor (β)	ARGMIN	MVH	SIKH
0.1	8.4	4.1	0.3
0.3	26.4	12.7	2.2
0.5	36.9	20.3	2.6
0.7	47.7	24.2	4.2
0.9	37.8	17.7	3.6

Table 3: Effect of the overtime cost on the relative performance of the heuristics (% deviation from the MLKH heuristic solution)

Heuristic SIKH, which requires solving a 0-1 problem to optimality, was the closest to MLKH but never did it generate a better solution. The other heuristics are generally much worse than MLKH regardless of the problem structure and size. The poor quality of the ARGMIN solution can be viewed as an indication of the level of difficulty of the problem. Obviously, when the overtime cost is negligible, the optimal solution is very close to the ARGMIN solution and the problem is easy. Otherwise, the problem of finding a good heuristic solution is very difficult. The gap between MLKH and the remaining heuristics appears to become larger for higher overtime costs, i.e., as the problem becomes harder.

We now evaluate MLKH intrinsically by comparing the cost of the solution provided by MLKH to the value of the lower bound obtained by solving the 0-1 surrogate relaxation problem generated at the termination of ML. The lower bounding procedure ML, and the upper bounding procedure MLKH are closely related, since both use multipliers derived by ML. Better multipliers lead both to tigher lower bounds and better heuristic solutions. We compare our heuristic to a lower bound rather than the optimal solution because of the difficulty of solving large and dense 0-1 problems to optimality using existing commercial codes. The lower bound, however, is obtained by solving a single constraint 0-1 multiple choice problem for which we coded a specialized B&B procedure. Ten problems are generated for each set of problem parameters. In Table 4, we report the mean and standard deviation of the relative deviation of the heuristic solution value from the respective lower bound value.

The average gap appears to increase with the number of alternatives and decrease with the number of items. The heuristic considers only minor modifications of the (continuous) solution obtained from the ML procedure, and therefore investigates fewer of the combinations of schedules as the number of alternatives increases. The decrease in the average gap with the number of items is expected, since increasing the number of suppliers makes it easier to fit the schedules together. The overtime cost and the number of side constraints also influence the performance of the heuristic. As the number of side constraints increases, the more difficult it becomes to achieve a good one-dimensional representation of the problem. The same applies to the effect of the β factor. As β increases, the scheduling problem becomes much more difficult. The schedules need to fit together very well in order to avoid exceeding the regular capacity. Both the number of side constraints and the β factor would be expected to affect the duality gap.

S	K	H	β	Average Gap	Maximum Gap	Gap Std. Dev.
25	10	5	0.5	3.4	7.3	1.9
	20			3.7	7.9	2.2
	30			3.9	7.6	1.9
	40			4.2	10.0	2.3
10	25	5	0.5	4.2	7.0	1.1
20				3.8	6.3	1.1
30				3.8	6.8	1.7
40				3.1	7.5	0.7
40	25	5	0.1	1.6	3.4	0.9
			0.3	2.4	5.0	1.2
			0.5	3.1	7.5	0.7
			0.7	3.9	10.0	2.3
			0.9	4.8	9.2	2.7
40	20	3	0.5	0.5	1.0	0.2
		4		1.8	3.3	0.8
		5		3.0	5.0	1.1
		6		4.4	7.3	2.2
		8		4.7	6.9	1.7
		10		4.9	8.6	2.0

Table 4: Effect of the different factors on the percentage gap between the MLKH heuristic solution and the lower bound obtained by ML.

Although we have observed some trends, the results in Table 4 should be analyzed with caution. A major portion of the deviations may be due to the duality gap rather than to the gap between the heuristic solution and the optimal solution, and the gap would be expected to increase with problem difficulty. For these problems, the heuristic was, on the average, less than 5% from the lower bound, which represents very good overall perfromance. The low standard deviations suggest that the heuristic was consistent in its performance for each case analyzed.

The maximum reported deviation is 10%. Among the 80 problems generated in the first experiment (variable problem size, β=0.5, H=5) three out of four have a deviation less than 5%

and less than one tenth have a deviation greater than 7%. Even for cases where the heuristic did not appear to perform well (β=0.9, H=6,8 and 10) only five of the forty problems resulted in a deviation greater than 7%, and more than half of the problems were below the 5% mark.

Computational tractability:

ML is an iterative procedure in which another iterative procedure, SL (or SLS) is imbedded. At each iteration of SL, a continuous one-dimensional knapsack problem is solved which can be solved in linear time. The total number of procedure ML iterations and the number of LPs solved at each iteration determine the computation time required. Although it is difficult to derive a theoretical bound, the results of our tests suggest that the number of iterations of procedure ML barely exceeds the number of multipliers (H). Figure 1 shows a frequency graph of the number of iterations needed beyond H. The experiment consisted of 60 randomly generated problems. There is a set of ten problems for each value of H (= 3, 4, 5, 6 ,8 and 10.)

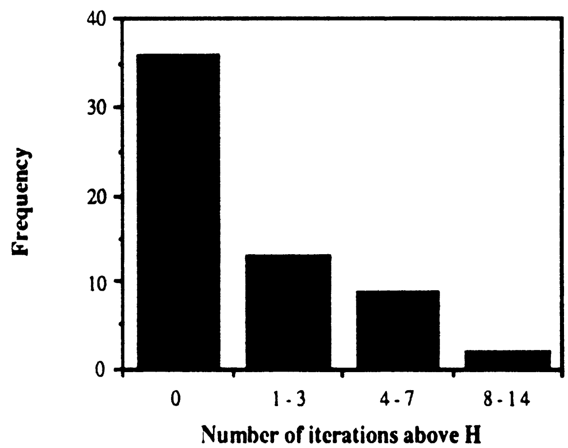

Figure 1: Number of iterations required by procedure ML in 60 randomly generated problems (iterations above H, the number of multipliers)

At each iteration of procedure ML, we find optimal multipliers for a two-constraint problem. We solved 130 problems with different values of S, K, and H. The average number of LPs solved at each iteration of procedure ML is reported in a frequency graph in Figure 2. The results did not suggest any dependency on the problem size. The spread of values depicted in Figure 6.4 is illustrative of the variances that are found within a set of problems of a given size. The total number of LPs solved remains manageable, around one hundred in most cases.

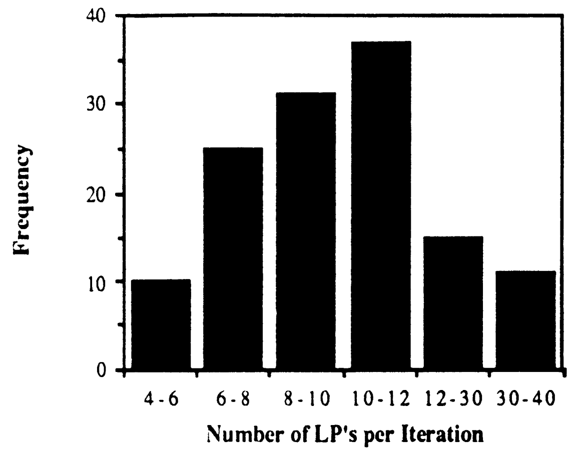

Figure 2: Average number of LP's solved in one iteration of ML.
(values for 130 randomly generated problems)

Procedure MLKH consists of "rounding" the fractional solution obtained each time a surrogate problem is solved in ML. As discussed in section 4, the additional computational effort for this is linear in the total number of alternatives and in the number of side constraints. In Table 5, we report the total CPU time for all routines. Since our code is only experimental, these values can be considerably reduced, for example, by implementing a linear time algorithm to solve the LPs, which in our code is done in a polynomial time. The sorting operations, which consume 20 to 30% of the computation time, can also be executed much faster.

S	K	H	β	Average CPU time	Standard deviation
25	10	5	0.5	20.5	10.0
	20			21.6	7.2
	30			51.7	24.1
	40			69.4	40.8
10	25	5	0.5	22.0	12.1
20				34.2	18.6
30				42.5	19.2
40				63.7	41.9
40	20	3	0.5	34.4	5.3
		4		29.3	12.6
		5		54.0	31.7
		6		53.2	25.7
		8		56.1	11.4
		10		72.6	15.2

Table 5: Average CPU times and standard deviations (in secs.) for different data sets.

The average CPU times in Table 5 suggest that the problem size has little effect on the computation time. It is clear that the solution times are very reasonable considering the sophistication of our algorithm (solution of dozens of LPs, execution of a B&B algorithm) and the quality of the solution obtained for this difficult problem.

6. Conclusions

We address a multi-item scheduling problem, where we must choose among a small number alternative schedules for dozens of items. We seek to coordinate the production schedules over several days so as to smooth the workload, thereby controlling overtime. For each item, the total production quantity in each alternative schedule is the same. However the schedules differ in both their workload profiles and in their costs. If the emphasis is on production costs, which usually exhibits economies of scale because of setups, then schedules with large production batches will be more economical. On the other hand, if reducing inventory is the main focus then just-in-time production schedules may be more appropriate. The overtime cost may also be an important factor in choosing the best schedule for each item while considering all the items simultaneously.

The scheduling problem is modeled as a multidimensional multiple-choice knapsack problem in which the we allow the knapsacks to be enlarged at some cost. It is difficult to solve the problem to optimality. We show that simplistic approaches provide poor solutions, and propose a heuristic which is based on the solution to a surrogate relaxation to the problem. The multidimensional problem is transformed to a one-dimensional problem using surrogate multipliers. The multipliers are derived by an iterative procedure which requires solving dozens of one-dimensional continuous knapsack problems. The solution to these LP's is appropriately rounded to provide a good solution to the original problem.

Computational results show that the bound from the surrogate relaxationis very tight. The heuristic solution is, in general, less than 5% from this bound. The suggested algorithm derives good surrogate multipliers that are used to generate both a good heuristic solution and a tight lower bound. The algorithm is computationally tractable for reasonably large problems. For a mainframe computer, the computation times will range in the milliseconds for problems with 40 items, 25 alternatives each, and a 7-day horizon.

The lower and upper bounding procedures proposed in this paper can be used to develop efficient B&B methods for solving the problem to optimality. Bounds could be based on surrogate relaxation, as suggested here. The initial multipliers at a given node may be set at the terminal values obtained at the predecessor node in the B&B tree, which will help to speed convergence. The branching scheme may also be based on the surrogate relaxation, which if solved as a LP, will have at most two fractional values. Dichotomy rules of the type used for the one-dimensional multiple choice problem can be used for branching.

When overtime is permitted, feasibility is not an issue and the solutions obtained by our heuristic are good enough to disregard solving the problem optimally. On the other hand, if the resources are fixed or if the maximum amount of overtime is restricted, then our heuristic solution may be infeasible. Most of the steps to derive the multipliers can be easily adapted to handle these cases. The multipliers can still be used in different heuristic procedures that ensure feasibility or in a B&B scheme to solve the problem to optimality.

In conclusion, we have developed an algorithm which can be applied to any multiple-choice problem with a few knapsack type side-constraints. We suggested an efficient procedure to derive good heuristic solutions, and tight lower bounds to assess the quality of these solutions. We have focused in our experiments on scheduling problems. Further work is needed to test the approach on other applications.

ACKNOWLEDGEMENT

The authors are grateful to a major U.S. automobile manufacturer for funding through a research contract to the University of Michigan.

BIBLIOGRAPHY

Bean, J.C. (1987), "Multiple Choice Knapsack Functions," *Technical Report 87-26*, Industrial and Operations Engineering Department, The University of Michigan, Ann Arbor, Mi.

Ben-Kheder, N. and Yano, C.A. (1990), ' The Multi-Supplier Delivery Scheduling Problem: Heuristics and Lower Bounds", *Technical Report 90-4*, Industrial and Operations Engineering Department, The University of Michigan, Ann Arbor, Mi.

Bitran, G.R.and Matsuo, H. (1986), "The Multi-Item Capacitated Lot-Size Problem: Error Bounds of Manne's Formulations," *Management Science* **32**, pp. 350-358.

Dudzinski, K. (1984), "A Dynamic Programming Approach for Solving the Multiple Choice Knapsack Problem," *Bulletin of the Polish Academy of Science, Technical Science* **32**, pp. 325-332.

---------------- and Walukiewicz, S. (1984), "A Fast Algorithm for the Linear Multiple Choice Knapsack Problem," *Operations Research Letters* **3**, pp. 205-209.

--- (1987), "Exact Methods for the Knapsack Problem and its Generalizations." *European Journal of Operational Research* **29**, pp. 3-21.

Dyer, M.E. (1984), "An O(n) Algorithm for the Multiple Choice Knapsack Linear Program," *Mathematical Programming* **29**, pp. 57-63.

-----------, Kayal, N. and Walker, J. (1984), "A Branch-and-Bound Algorithm for Solving the Multiple Choice Knapsack Problem," *Journal of Computational and Applied Mathematics* **11**, pp. 231-249.

Dzielenski, B.P. and Gomory, R.E. (1963), "Optimal Programming of Lot-Sizes, Inventory and Labor Allocations," *Management Science* **11**, pp. 874-890.

Gavish, B., and Pirkul, H., "Efficient Algorithms for Solving Multiconstraint Zero-One Knapsack Problems to Optimality," *Mathematical Programming* **19**, pp.255-278.

Gilmore, P.C. and Gomory, R.E. (1966), "The Theory of Computation of Knapsack Functions," *Operations Research* **14**, pp. 1045-1074.

Glover, F. (1975), "Surrogate Constraint Duality in Mathematical Programming," *Operations Research* **23**, pp. 434-451.

Karwan, M.H. and Rardin, R.L. (1979), "Some Relationships between Lagrangian and Surrogate Duality in Integer Linear Programming," *Mathematical Programming* **17**, pp. 320-334.

--- (1984), "Surrogate Dual Multiplier Search Procedures in Integer Programming," *Operations Research* **32**, pp.52-69.

Lasdon, L.S. and Terjung, R.C. (1971), "An efficient Algorithm for Multi-Item Scheduling," *Operations Research* **19**, pp. 946-969.

Loulou, R. and Michaelides, E. (1979), "New Greedy Like Heuristics for the Multidimensional 0-1 Knapsack Problem," *Operations Research* **27**, pp. 1101-1114.

Manne, A.S. (1958), "Programming of Economic Lot-Sizes," *Management Science* **4**, pp. 115-135.

Nauss, R.M. (1979), "The 0-1 Knapsack Problem with Multiple Choice Constraints," *European Journal of Operational Research* **2**, pp. 125-131.

Narasimhan, S. and Pirkul, H. (1986), "Efficient Algorithms for the Multiconstraint General Knapsack Problem," *IIE Transactions* **18**, pp. 195-203.

Pirkul, H. (1987), "A Heuristic Solution Procedure for the Multiconstraint Zero-One Knapsack Problem," *Naval Research Logistics* **34**, pp. 161-172.

Shih, W. (1979), "A Branch and Bound Method for the Multiconstraint Zero-One Knapsack Problem," *Journal of the Operational Research Society* **30**, pp. 369-378.

Sinha, P. and Zoltners, A.A. (1979), "The Multiple Choice Knapsack Problem", *Operations Research* **27**, pp. 503-515.

Toyoda, Y. (1975), "A Simplified Algorithm for Obtaining Approximate Solutions to Zero-One Programming Problems," *Management Science* **21**, pp. 1417-1427.

Weingartner, H.M. and Ness, D. N. (1967), "Methods for the Solution to the Multidimensional 0/1 Knapsack Problem," *Operations Research* **15**, pp. 83-103.

Zemel, E. (1984) "An O(n) Algorithm for the Linear Multiple Choice Knapsack Problem and Related Problems," *Information Processing Letters* **18**, pp. 132-138.

Appendix

Theorems and Proofs

Lemma 1: Let $\mu < W_2/W_1$. Then an optimal solution to SP_μ has $V_2=0$. Similarly, if $\mu > W_2/W_1$ then an optimal solution to SP_μ has $V_1=0$.

Proof: We prove the result for the case where $\mu < W_2/W_1$. The proof is similar for the case $\mu > W_2/W_1$.

Let (Y,V_1^*,V_2^*) be an optimal solution to $SP\mu$ and suppose that $V_2^*>0$. Let $V'_1=V_1^*+\mu V_2^*$. Then

$$(L_1 + \mu L_2) Y^* \le (R_1 + \mu R_2) + (V_1^* + \mu V_2^*)$$
$$\le (R_1 + \mu R_2) + V_1'.$$

Therefore $(Y^*,V'_1,0)$ is a feasible solution to $SP\mu$. Now since $\mu < W_2/W_1$ and $V^*_2 > 0$, we have

$$CY^* + W_1V_1' = CY^* + W_1V_1^*+W_1\mu V_2^*$$
$$< CY^* + W_1 V_1^*+ W_2 V_2^*$$

which contradicts the optimality of (Y^*,V_1^*,V_2^*), for $V_2^* > 0$.

In the following, for simplicity, we will assume that $\mu < W_2/W_1$. Similar results hold for the case $\mu>W_2/W_1$ (simply reindex the constraints). Based on this assumption and the results of Lemma 1, V_2 is equal to zero and we write the solution to $SP\mu$ as (Y,v), where v is equal to V_1.

Lemma 2: Let (Y,v) be an optimal solution to $SP\mu$. Then at most one constraint is violated by this solution.

Proof: Let (Y,v) be an optimal solution to $SP\mu$. Assume that both constraints of problem P2 are violated by this solution, i.e., $L_1 Y > R_1 + v$ and $L_2 Y > R_2$. This implies that, since $\mu \ge 0$, we have $(L_1 + \mu L_2) > (R_1 + \mu R_2) + v$, which contradicts the feasibility of (Y,v) in problem $SP\mu$.

Lemma 3: **Let (Y,v) be an optimal solution to $SP\mu$ that does not satisfy constraint (1) of P2. Then $\mu^* \le \mu$, where μ^* denotes the optimal multiplier.**

Proof: Let (Y,v) be an optimal solution to $SP\mu$ that does not satisfy the first constraint of problem P2, i.e.,

$$L_1Y > R_1 . \qquad (A1)$$

If $\mu=\mu^*$, i.e., $z(SP\mu) = \operatorname{Sup}_{\gamma>0} SP\gamma$, then we are done. Otherwise, we show that if the value of μ is increased by a positive increment η, then the surrogate bound decreases. As a consequence μ^*

must be smaller than μ. Since the first constraint is violated, we know by Lemma 2 that the second one is not, i.e.,

$$L_2Y \leq R_2 \tag{A2}$$

and since (Y,v) is feasible to SPμ, we have:

$$(L_1+\mu L_2) \leq (R_1+\mu R_2) + v \tag{A3}$$

Now let $\mu'=\mu+\eta$, where $\eta>0$. Then

$$\begin{aligned}(L_1 + \mu' L_2)Y &= (L_1 + \mu L_2)Y + \eta L_2 \\ &\leq (L_1 + \mu L_2)Y + \eta R_2 && \text{(by A2)} \\ &\leq (R_1 + \mu R_2) + \eta R_2 + v && \text{(by A3)} \\ &= (R_1 + \mu' R_2) + v\end{aligned}$$

We conclude that (Y,v) is also feasible to SPμ'. Therefore the set of feasible solution to SPμ is included in the set of feasible solutions to SPμ'. Hence $z(SP\mu') \leq z(SP\mu)$. Since μ is not optimal then neither is μ', $\mu'\geq\mu$.

<u>Lemma 4</u>: Let (Y,v) be an optimal solution to SP_μ that does not satisfy constraint (2) of P2. Then $\mu^* \geq \mu$.

<u>Proof</u>: Let (Y,v) be an optimal solution to SPμ that does not satisfy the second constraint of problem P2, i.e., $L_2Y > R_2$. (A4)

If $\mu=\mu^*$, i.e., $z(SP\mu) = \text{Sup}_{\gamma>0} SP\gamma$, then we are done. Otherwise, we show that if the value of μ is decreased by a positive increment η, then the surrogate bound decreases. As a consequence μ^* must be higher than μ. Since (Y,v) is feasible to SPμ, we have:

$$(L_1+\mu L_2) \leq (R_1+\mu R_2) + v \tag{A5}$$

Now let $\mu'=\mu-\eta$, where $\eta>0$. Then

$$\begin{aligned}(L_1 + \mu' L_2)\, Y &= (L_1 + \mu L_2)\, Y - \eta L_2 \\ &\leq (L_1 + \mu L_2)Y - \eta R_2 && \text{(by A4)} \\ &\leq (R_1 + \mu R_2) - \eta R_2 + v && \text{(by A5)} \\ &= (R_1 + \mu' R_2) + v\end{aligned}$$

We conclude that (Y,v) is also feasible to SPμ'. Therefore the set of feasible solution to SPμ is included in the set of feasible solutions to SPμ'. Hence $z(SP\mu') \leq z(SP\mu)$. Since μ is not optimal then neither is μ', $\mu'\leq \mu$.