MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO DE PESQUISAS ESPACIAIS

AUTORIZAÇÃO PARA PUBLICAÇÃO
AUTHORIZATION FOR PUBLICATION

PALAVRAS CHAVES/KEY WORDS

AUTORES AUTHORS

SURROGATE
0-1 MULTIKNAPSACK
LAGRANGEAN RELAXATIONS

AUTORIZADA POR/AUTHORIZED BY

Luiz A. Vieira Dias
Chairman of LAC

AUTOR RESPONSAVEL
RESPONSIBLE AUTHOR

Luiz A.N. Lorena

DISTRIBUIÇÃO/DISTRIBUTION

- [ ] INTERNA / INTERNAL
- [x] EXTERNA / EXTERNAL
- [ ] RESTRITA / RESTRICTED

REVISADA POR / REVISED BY

CDU/UDC

519.87

DATA / DATE

Outubro/1990

PUBLICAÇÃO Nº
PUBLICATION NO

INPE-5160-RPE/635

ORIGEM
ORIGIN

LAC

PROJETO
PROJECT

POPES

Nº DE PAG.
NO OF PAGES

25

ULTIMA PAG.
LAST PAGE

20

VERSÃO
VERSION

Nº DE MAPAS
NO OF MAPS

TÍTULO/TITLE

A MONOTONE DECREASING ALGORITHM FOR THE 0-1 MULTIKNAPSACK DUAL PROBLEM

AUTORES/AUTHORSHIP

Luiz Antonio Nogueira Lorena
G. Plateau*

RESUMO-NOTAS/ABSTRACT-NOTES

We show in this work a monotone decreasing algorithm for the 0-1 Multiknapsack dual problem. The subgradient type algorithm solves at each iteration a Continuous Surrogate relaxation, and a simple control at the subgradient updating produces a monotone decreasing sequence of Lagrangean relaxations. A lot of computational tests with problems of the literature are presented.

OBSERVAÇÕES/REMARKS

Publicated by: Department de Mathematique et Informatique - Université PARIS-NORD/França - Prépublication 89-1, Janvier 1989.

RESUMO

Mostra-se nesse trabalho um algoritmo que produz uma sequência monotonicamente decrescente de valores duais para o problema multidimensional da mochila em variáveis 0-1. No algoritmo, tipo subgradiente, resolve-se a cada iteração uma relaxação "surrogate" contínua, e um controle simples na atualização do subgradiente produz uma sequência monotonicamente decrescente de relaxações Lagrangeanas. São apresentados vários testes computacionais com problemas da literatura.

CONTENTS

A MONOTONE DECREASING ALGORITHM FOR THE 0-1 MULTIKNAPSACK DUAL PROBLEM

L.A.N.LORENA(*) and G.PLATEAU(**)

Abstract : We show in this work a monotone decreasing algorithm for the 0-1 Multiknapsack dual problem. The subgradient type algorithm solves at each iteration a Continuous Surrogate relaxation, and a simple control at the subgradient updating produces a monotone decreasing sequence of Lagrangean relaxations. A lot of computational tests with problems of the literature are presented.

Key words : 0-1 Multiknapsack, Surrogate and Lagrangean relaxations

---

(*) Instituto de Pesquisas Espaciais - INPE, Ministério de Ciência e Tecnologia, Caixa Postal 515, 12201 São José dos Campos-SP, Brazil.
(**) LIPN - Université de PARIS-NORD, CSP, Département de Mathématiques et Informatique, Av.J.B.Clément, 93430 Villetaneuse, France.

1. INTRODUCTION:

In this work we show a monotone decreasing algorithm for the dual of the 0-1 Multiknapsack problem.

The 0-1 Multiknapsack problem can be defined as

$$
(P) \qquad \begin{array}{ll} \max & cx \\ \text{s.t.} & Ax \leq b \\ & x \in \{0,1\}^n, \end{array}
$$

where, $c \in N^n$, $b \in N^m$, A is a $m \times n$ dense non--negative integer matrix and $\{0,1\}^n = \{x \in R^n : x_j = 0 \text{ or } x_j = 1;\ j = 1, \ldots, n\}$.

In section 2 we define the Lagrangean and Surrogate Continuous relaxations, and show one condition for the equality between there optimal values. The algorithm for solving the dual of (P) is presented in section 3, and analysed in section 4. In section 5 we present computational tests for 13 problems of the literature.

2. LAGRANGEAN AND SURROGATE CONTINUOUS RELAXATIONS

In this section, we show one condition for the equality between the optimal values of the Lagrangean and the Surrogate Continuous relaxations of (P).

The Surrogate Continuous relaxation can be defined as

$$
(SR_w) \qquad \begin{array}{ll} \max & cx \\ \text{s.t.} & wAx \le wb \\ & x \in [0,1]^n \end{array}
$$

where $w \in R_+{}^m$,
and $x \in [0,1]^n$ means $0 \le x_j \le 1; j=1,\ldots,n$.

And the Lagrangean relaxation as

$$
(LR_w) \qquad \begin{array}{l} \max \{cx - w(Ax - b)\}, \\ \text{s.t } x \in \{0,1\}^n \end{array}
$$

where $w \in R_+{}^m$.

The solution of (SR_w) is obtained as follows (Dantzig [1]):

- sort the ratios c_j/wA^j in decreasing order,
- fix variables at 1 according this order until the infeasibility of the constraint.

Let i* be the variable index that makes the infeasibility of the constraint; x_{i*} is the basic variable, and let $J_w = \{j \in \{1, 2, ..., n\} : x_j = 1$ in the solution of $(SR_w)\}$. Then,

$$v(SR_w) = \sum_{j \in J_w} c_j + c_{i*} x_{i*},$$

$$\text{where } x_{i*} = w(b - \sum_{j \in J_w} A^j)/wA^{i*}$$

$$v(SR_w) = \sum_{j \in J_w} c_j - c_{i*} \frac{wg_w}{wA^{i*}}, \quad \text{for } g_w = \sum_{j \in J_w} A^j - b.$$

$$\text{Where } wg_w \leq 0 \text{ and } |wg_w| \leq wA^{i*}.$$

$$\text{Let } \lambda_w = c_{i*}/wA^{i*},$$

that is the solution of the dual problem of the Surrogate Relaxation of (P). We show in the following a sufficient condition for the equality between the optimal values of the Surrogate Continuous and the Lagrangean relaxation of (P).

PROPOSITION 1: If $w_\lambda = \lambda_w w$ then $v(SR_w) = v(LRw_\lambda)$.

Proof:
$$v(LRw_\lambda) = \max_{x \in \{0,1\}^n} \{cx - w_\lambda(Ax - b)\} =$$

$$= \max_{x \in \{0,1\}^n} \{cx - \lambda_w w(Ax - b)\} =$$

$$= \max_{x\in\{0,1\}^n} \{cx - (c_{i*}/wA^{i*})w(Ax - b)\} =$$

$$= \max_{x\in\{0,1\}^n} \{\sum_{j=1}^{n} [c_j - (c_{i*}/wA^{i*})wA^j]x_j + (c_{i*}/wA^{i*})wb\} =$$

$$= \sum_{j\in J_w} c_j - c_{i*}\frac{wg_w}{wA^{i*}} = v(SR_w).$$

This result shows that if we know an optimal solution of the problem (SR_w), then there is a Lagrangean Relaxation (LRw_λ) with the same optimal value. The optimal solution of (LRw_λ) will be:

$$\begin{cases} x_j = 1, & \text{for all } j\in J_w, \\ x_j = 0, & \text{otherwise.} \end{cases}$$

COROLLARY 1: Given a multiplier $w \in R_+^m$, and (from the solution of (SR_w)) $i*$, λ_w and J_w; for any set $J'_w \subset \{1, 2, \ldots, n\}$ such that $wg'_w \leq 0$, $g'_w = \sum_{j\in J'_w} A^j - b$, and any $i' \in \{1, 2, \ldots, n\} - J'_w$, we have

$$\sum_{j\in J'_w} c_j - \lambda'_w wg'_w + \sum_{j\in K} |c_j - \lambda'_w wA^j| \leq v(SR_w),$$

where $\lambda'_w = \dfrac{c_{i'}}{wA^{i'}}$, and

$$K = (J_w \cup J'_w) - (J_w \cap J'_w).$$

Proof: Suppose that $J_w = J'_w$, then $K = \emptyset$ and $g'_w = g_w$:

- if $i* = i'$, then $\lambda'_w = \lambda_w$, and

$$\sum_{j \in J'_w} c_j - \lambda'_w w g'_w = v(SR_w);$$

- if $i* \neq i'$, then

$$\sum_{j \in J'_w} c_j - \lambda'_w w g'_w \leq \sum_{j \in J_w} c_j - \lambda_w w g_w = v(SR_w).$$

Suppose now that $J_w \neq J'_w$, then

- if $i* = i'$, then $\lambda'_w = \lambda_w$ and

$$\sum_{j \in J'_w} c_j - \lambda_w w g'_w + \sum_{j \in K} |c_j - \lambda_w w A^j| =$$

$$= \sum_{j \in J'_w} (c_j - \lambda_w w A^j) + \lambda_w w b - \sum_{j \in K_{J'}} (c_j - \lambda_w w A^j) +$$

$$\sum_{j \in K_J} (c_j - \lambda_w w A^j) = \sum_{j \in J_w} c_j - \lambda_w w g_w = v(SR_w),$$

where $K_{J'} = J'_w - (J_w \cap J'_w)$ and
$K_J = J_w - (J_w \cap J'_w)$;

- if $i* \neq i'$, then

$$\sum_{j \in J'_w} c_j - \lambda'_w w g'_w + \sum_{j \in K} |c_j - \lambda'_w A^j| =$$

$$\sum_{j \in J'_w} (c_j - \lambda'_w w A^j) + \lambda'_w w b - \sum_{j \in K_{J'}} (c_j - \lambda' w A^j) +$$

$$\sum_{j \in K_J} (c_j - \lambda'_w w A^j) = \sum_{j \in J_w} c_j - \lambda'_w w g_w \leq v(SRw).$$

This result will be used in section 4 for the analysis of the algorithm given in the next section.

3. ALGORITHM:

{ For any Surrogate Relaxation of (P) (SR_w),

let we denote (see section 2)

i* the index of the basic variable

$\lambda_w = c_{i*} / wA^{i*}$ and $g_w = \sum_{j \in J_w} A^j - b$ }

Compute $w \in R^m_+$; solve (SR_w); $\theta \leftarrow +\infty$;

while $v(SR_w) < \theta$ **do**

$\theta \leftarrow v(SR_w)$;

compute $t \in R_+$;

$$w \leftarrow w + \frac{t}{\lambda_w \|g_w\|^2} g_w;$$

for each i in {1,...,m} such that $w_i < 0$ **do**

$w_i \leftarrow 0$;

endfor;

solve (SR_w);

endwhile

Comments: (i) The current problem (SR_w) is solved obtaining λ_w and J_w , that is, by proposition 1 we also obtain the solution of (LRw_λ) for $w_\lambda = \lambda_w w$. The problem (SR_w) can be solved by the expected linear time complexity algorithm **NKR** of Fayard and Plateau [2].

(ii) For the stop criteria, we suppose that the sequence of values $v(SR_w)$ (or $v(LRw_\lambda)$) is monotonous decreasing and we stop when the minimum is reached. In the next section is showed one condition for that.

(iii) Given $\delta_w(t) = \dfrac{t}{\lambda_w \|g_w\|^2}$ the following definitions for t can be used:

(a) $t_1 = [v(SR_w) - v_h]/p$, where v_h is a lower bound on $v(P)$ obtained by any heuristics, and $p > 0$ is used in the computational tests of section 5, to make

$$10^{-4} \leq \delta_w(t_1) \leq 10^{-3},$$

with the initial w of the algorithm;

(b) $t_2 = [v(SR_w) - \sum_{j\in J_w} c_j]/p = -(\lambda_w/p)wg_w$,

where p is defined as in (a). In this particular case, if $p = 1$, at each iteration the updated value of w is $w - (wg_w/\|g_w\|^2)g_w$, that is, the difference between w and its projection on g_w;

(c) t_3 is assigned to a positive constant value defined to make $10^{-4} \leq \delta_w(t_3) \leq 10^{-3}$, with the initial w of the algorithm.

4. ANALYSIS OF THE ALGORITHM

Let w and w' two consecutive iterations of the algorithm. First, let we look $v(SR_{w'})$.

PROPOSITION 2:

$$v(LRw'_\lambda) = v(SR_{w'}) = \sum_{j \in J_{w'}} c_j - c_{i*'} \frac{[wg_{w'} + \delta_w(t) g_w g_{w'}]}{[wA^{i*'} + \delta_w(t) g_w A^{i*'}]}.$$

Proof: Similar to proposition 1, by noting that

$w'_\lambda = \lambda_{w'} w'$, $\lambda_{w'} = \dfrac{c_{i*'}}{w'A^{i*'}}$, $w' = w + \delta_w(t) g_w$, $i*'$ is the basic variable of $(SR_{w'})$ and $J_{w'} = \{j \in \{1,\dots,n\}: x_j = 1$ in the optimal solution of $(SR_{w'})\}$.

COROLLARY 2: (i) $\delta_w(t) < \max_{j \in \{1,\dots,n\}} \{ wA^j / |g_w A^j| \}$;

(ii) if $\delta_w(t) \leq |wg_{w'}| / |g_w g_{w'}|$ then $wg_{w'} \leq 0$.

Proof: From the solution of $(SR_{w'})$ we know that

(i) $w'A^{i*'} > 0$

$wA^{i*'} + \delta_w(t) g_w A^{i*'} > 0$ and then

$\delta_w(t) < wA^{i*'} / |g_w A^{i*'}|$ or $\delta_w(t) < \max_{j \in \{1,\dots,n\}} \{ wA^j / |g_w A^j| \}$

(ii) $w'g_{w'} \leq 0$

$wg_{w'} + \delta_w(t) g_w g_{w'} \leq 0$

and if $\delta_w(t) \leq |wg_{w'}| / |g_w g_{w'}|$ then $wg_{w'} \leq 0$.

This result shows that we always have an upper bound for $\delta_w(t)$ at each iteration of the algorithm, and for $\delta_w(t)$ "sufficiently small" $wg_{w'} \leq 0$.

Let us define (see section 2)

$$\Omega = \sum_{j \in K} |c_j - \lambda'_w w A^j| ,$$

where $K = (J_w \cup J_{w'}) - (J_w \cap J_{w'})$, and

$$\lambda'_w = \frac{c_{i*'}}{wA^{i*'}}.$$

COROLLARY 3: If $wg_{w'} \leq 0$ then

$$v(LRw_\lambda) = v(SR_w) \geq \sum_{j \in J_{w'}} c_j - \lambda'_w w g_{w'} + \Omega .$$

Proof: immediate from corollaries 1,2 and proposition 2. ∎

In the next proposition we show how to make a control of the parameter t (and as a consequence of $\bar{\delta}_w(t)$) for that $v(SR_{w'}) \leq v(SR_w)$ $(v(LRw'_\lambda) \leq v(LRw_\lambda))$.

PROPOSITION 3: If $J_w \neq J_{w'}$ and $\bar{\delta}_w(t)$ is "sufficiently small" then $v(SR_{w'}) \leq v(SR_w)$.

Proof: For $\bar{\delta}_w(t)$ "sufficiently small",

$$\left| c_{i*'} \frac{w' g_{w'}}{w' A^{i*'}} - c_{i*'} \frac{w g_{w'}}{w A^{i*'}} \right| \leq \Omega ,$$

then due to proposition 2,

$$v(SR_{w'}) \leq \sum_{j \in J_{w'}} c_j - c_{i*'} \frac{w g_{w'}}{w A^{i*'}} + \Omega ,$$

and then due to corollaries 2 and 3, $v(SR_{w'}) \leq v(SR_w)$.

Remark: In the next section we make some computational tests to validate this proposition and to search for the best value of $\delta_w(t)$ (at the initial iteration of the algorithm) when the algorithm of section 3 is applyed to problems of the literature.

In the next proposition we show that if $J_w = J_{w'}$ then $v(SR_w) > v(SR_{w'})$ for $t > 0$.

PROPOSITION 4: If $J_w = J_{w'}$ then

(i) $t \le \lambda_w \; |wg_w|$;

(ii) $t \le c_{i*}$;

(iii) $t < \lambda_w \|g_w\|^2 wA^{i*'} / \; |g_w A^{i*'}|$;

(iv) $0 \le v(SR_w) - v(SR_{w'}) \le \lambda_w \; |wg_w|$, for $0 \le t \le \lambda_w \; |wg_w|$.

Proof: If $J_w = J_{w'}$, then $g_w = g_{w'}$ and

(i) $w'g_{w'} = (w + \delta_w(t) g_w) g_w = wg_w + t/\lambda_w$,but $w'g_{w'} \le 0$, then $wg_w + t/\lambda_w \le 0$ or $t \le \lambda_w \; |w \, g_w|$.

(ii) from the solution of (SR_w) $|wg_w| \le w \; A^{i*}$, then $t \le c_{i*}$

(iii) from the solution of $(SR_{w'})$ we know that $w'A^{i*'} > 0$, or $wA^{i*'} + \delta_w(t) g_w A^{i*'} > 0$

$$wA^{i*'} + (t/\lambda_w \; \|g_w\|^2) g_w A^{i*'} > 0$$

$$t < \lambda_w \|g_w\|^2 wA^{i*'} / |g_w A^{i*'}|.$$

(iv) $$v(SR_w) - v(SR_{w'}) =$$

$$- \lambda_w w g_w + c_{i*'} \cdot \frac{[w g_w + t/\lambda_w]}{[wA^{i*'} + \delta_w(t) g_w A^{i*'}]} ,$$

and then from (i) and (iii)

for $t = \lambda_w |wg_w|$, $v(SR_w) - v(SR_{w'}) = - \lambda_w w g_w \geq 0$;

for $t = 0$, $v(SR_w) - v(SR_{w'}) = - \lambda_w w g_w + \lambda'_w w g_w \geq 0$, because $\lambda_w \geq \lambda'_w$.

5. COMPUTATIONAL TESTS:

In this section we present computational tests of applying the algorithm of section 3 to 13 problems of the literature. The main features of this tests are reported in table 1 and described above.

For the first column of table 1, the problems W1-W8 are of Weingartner and Ness [7], the problem F of Fleisher [3], the problems P6 and P7 of Petersen [5] and the problems ST1 and ST2 of Senju and Toyoda [6].

For each problem of column 1, the columns 2, 3 and 4 reports respectively the values of the optimal solution of (P) (v(P)), the linear programming relaxation of (P) (PL), and $v(SR_w{}^0)$, where $w^0{}_i = (\sum_{j=1}^{n} a_{ij} - b_i) / \sum_{j=1}^{n} a_{ij}$ $i=1,\dots,m$, is the initial value of the multiplier w used in the tests.

In the column 5 we have:

$k*$ = iteration number that algorithm stops, that is, when $v(SR_w) \geq v(SR_{w'})$ (step 2);

$v(SR_w{}^{k*})$ = the value of problem $(SR_w{}^{k*})$, or according proposition 1, the value of problem $(LRw^{k*}{}_\lambda)$ for $w^{k*}{}_\lambda = \lambda_w{}^{k*}w^{k*}$;

v_h = the lower bound on v(P), obtained by the following heuristic :

HEURISTIC: - sort the ratios c_j/wA^j, $j=1,\ldots,n$, in decreasing order (this is almost concluded for the solution of (SR_w);

- fix variables at 1 according this order while each constraint of (P) is feasible. When one constraint is violated, set the correspondent $x_j = 0$ and continue.

The value v_h is used in t of column 6, that reports $\delta_w(t)$ for $w = w^0$ and $w = w^{k*}$. For all tests of table 1, $t = t_1$ (of comment (iii-a)-section 3) is such that $10^{-4} \leq \delta_w{}^0(t) \leq 5.x10^{-3}$, that is "sufficiently small". For each problem of column 1, 4 values of $\delta_w{}^0(t)$ are tested to validate proposition 3. A direct observation of table 1 shows that when $\delta_w{}^0(t)$ increases $k*$ decreases, but generally the bound $v(SR_w{}^{k*})$ is worse in comparation with (PL).

The final column shows the ratios t/λ_w for $w = w^0$ and $w = w^{k*}$.

Because the sequence $\{v(SR_{w^{\cdot}})\}$ is monotone decreasing, the results of table 1 are obtained in very few iterations (k*) comparing with the traditional subgradient algorithms. The bounds $v(SR_w{}^{k*})$ are very good comparing with the (PL) bound (the solution of the (P) dual). The values of v_h are also very good (less than 1% error of v(P)).

6. CONCLUSION:

The good results of table 1 can be extended to other 0-1 problems for that problem (SR_w) is a continuous 0-1 knapsack problem, because in this case the propositions 1-4 remain valids.

We also have maded some tests using the definitions t_2 and t_3 (of comments in section 3). The results are comparables with of the table 1 for the bounds $v(SR_w{}^{k*})$, but with one increasing in k*.

The possibility of incorporate this algorithm with a reduction method of type described in Fréville and Plateau [2] is encouraging and in course of implementation.

REFERENCES:

[1] Dantzig G.B., "Discrete-variable extremum problems", Management Science 5(1957),266-277.

[2] Fayard D. and Plateau G., "Contribution a la résolution des programmes mathématiques en nombres entiers", dissertation, 1979.

[3] Fleisher J., Sigmap Newsletter 20, february 1976.

[4] Fréville A. and Plateau G., "Heuristics and reduction methods for multiple constraints 0-1 linear programming problems", European Journal of Operational Research 24(1986):206-215.

[5] Petersen C. C., "Computational experience with variant of the Balas algorithm applied to the selection of R&D projects", Management Science 13(9)(1967):736-750.

[6] Senju S. and Toyoda Y., "An approach to linear programming with 0-1 variables", Management Science 15(4)(1968):196-207.

[7] Weingartner H.M. and Ness D.N., "Methods for the solution of the multidimensional 0-1 Knapsack problem", Operations Research 15(1)(1967):83-103.

Prob.	v(P)	PL	$v(SRw^0)$	k* [for 4 values of $t(w = w^0)$]	$v(SR_w^{k*})$	v_h	$\delta_w^0(t)$	$\delta_w^{k*}(t)$	t/λ_w^0	t/λ_w^{k*}
W1 2x28	141278	142019.	148363.8	49	142019.	140618.	2.00×10^{-4}	4.27×10^{-4}	4.595	3.22×10^{-1}
				22	142019.	140618.	4.20×10^{-4}	9.39×10^{-4}	9.190	7.08×10^{-1}
				12	142019.	140618.	8.04×10^{-4}	1.80×10^{-3}	18.38	1.363
				1	142019.	140447.	4.02×10^{-3}	1.25×10^{-2}	91.90	9.447
W2 2x28	130883	131637.5	137840.2	54	131639.5	130723.	3.17×10^{-4}	3.47×10^{-5}	3.898	1.79×10^{-1}
				36	131638.4	130723.	4.75×10^{-4}	5.17×10^{-5}	5.848	2.66×10^{-1}
				17	131640.5	130723.	9.51×10^{-4}	1.06×10^{-4}	11.696	5.48×10^{-1}
				8	131648.2	130723.	1.90×10^{-3}	4.92×10^{-4}	23.393	1.12
W3 2x28	95677	99647.	102400.3	79	99647.6	95627.	3.81×10^{-4}	6.70×10^{-5}	1.503	1.87×10^{-1}
				66	99650.	95627.	4.57×10^{-4}	2.41×10^{-5}	1.804	2.64×10^{-1}
				32	99653.8	95627.	9.15×10^{-4}	4.82×10^{-5}	3.609	5.29×10^{-1}
				11	99651.7	95627.	2.28×10^{-3}	1.23×10^{-4}	9.023	1.354

Table 1

continue ... 1/5

Prob.	v(P)	PL	$v(SB_w^0)$	k* [for 4 values of t (w = w^0]	$v(SB_w^{k*})$ / v_h	$\delta_w^0(t)$ / $\delta_w^{k*}(t)$	t/λ_w^0 / t/λ_w^{k*}
W4 2x28	119337	122505.2	125896.	16	122508.2	2.18×10^{-4}	2.325
					119337.	2.46×10^{-4}	4.547
				10	122507.9	4.37×10^{-4}	4.651
					119337.	4.42×10^{-4}	8.153
				4	122595.5	8.75×10^{-4}	9.302
					119337.	4.16×10^{-4}	12.96
				1	123361.6	4.37×10^{-3}	46.512
					119337.	1.01×10^{-2}	100.479
W5 2x28	98796	100433.1	123271.1	67	100433.1	1.04×10^{-4}	9.521
					98796.	2.96×10^{-6}	3.38×10^{-1}
				34	100433.1	2.09×10^{-4}	19.043
					98796.	5.93×10^{-6}	6.76×10^{-1}
				17	100433.1	4.18×10^{-4}	38.089
					98796.	1.19×10^{-5}	1.336
				3	100433.1	2.09×10^{-3}	190.434
					98796.	7.11×10^{-5}	8.11
W6 2x28	130623	131335.	141157.1	48	131335.	2.71×10^{-4}	6.741
					130233.	2.44×10^{-5}	2.83×10^{-1}
				24	131335.	5.43×10^{-4}	13.483
					130233.	4.89×10^{-5}	5.67×10^{-1}
				16	131335.	7.76×10^{-4}	19.262
					130233.	6.98×10^{-5}	8.11×10^{-1}
				11	131335.	1.08×10^{-3}	26.967
					130233.	1.00×10^{-4}	1.167

continue ...

Prob.	v(P)	PL	$v(SR_w^0)$	k* [for 4 values of $t(w = w^0)$] / $v(SR_w^{k*})$ / v_h	$\bar{\delta}_w^0(t)$ / $\bar{\delta}_w^{k*}(t)$	t/λ_w^0 / t/λ_w^{k*}
W7 2x105	1095445	1095721.2	1101848.	20		
				1095725.	$1.07x10^{-4}$	8.208
				1095352.	$6.95x10^{-3}$	$4.08x10^{-1}$
				5		
				1095722.	$3.57x10^{-4}$	27.361
				1095352.	$1.61x10^{-3}$	1.45
				3		
				1095736.	$5.35x10^{-4}$	41.041
				1095352.	$6.05x10^{-4}$	2.42
				3		
				1095888.	$1.07x10^{-3}$	82.083
				1095352.	$1.00x10^{-3}$	6.43
W8 2x105	624319	628773.7	637939.9	14		
				628775.2	$3.58x10^{-4}$	3.418
				620060.	$3.04x10^{-4}$	1.62
				12		
				628777.	$4.78x10^{-4}$	4.558
				620060.	$3.97x10^{-4}$	2.119
				7		
				628777.	$7.17x10^{-4}$	6.837
				620060.	$1.26x10^{-3}$	3.261
				5		
				628780.2	$1.43x10^{-3}$	13.635
				620060.	$1.42x10^{-4}$	6.085
F 10x20	2139	2221.8	2447.8	20		
				2229.6	$3.35x10^{-4}$	33.76
				2068.	$4.58x10^{-4}$	10.805
				12		
				2231.	$4.46x10^{-4}$	45.02
				2068.	$3.20x10^{-4}$	7.7
				8		
				2235.8	$6.70x10^{-4}$	67.53
				2068.	$3.81x10^{-4}$	16.906
				5		
				2232.1	$1.34x10^{-3}$	135.06
				2068.	$1.42x10^{-4}$	37.778

Table 1

continue ...

Prob.	v(P)	PL	$v(SB_w^0)$	k* [for 4 values of t ($w = w^0$)]		
				$v(SB_w^{k*})$	$\bar{\rho}_w^0(t)$	t/λ_w^0
				v_h	$\bar{\rho}_w^{k*}(t)$	t/λ_w^{k*}
P6 5x39	10618	10672.3	11091.6	72		
				10676.1	3.79×10^{-4}	5.03
				10547.	1.93×10^{-5}	7.53×10^{-1}
				53		
				10675.9	5.05×10^{-4}	6.7
				10547.	2.57×10^{-5}	1.006
				35		
				10677.	7.58×10^{-4}	10.06
				10547.	3.97×10^{-5}	1.548
				16		
				10678.1	1.51×10^{-3}	20.12
				10547.	8.85×10^{-5}	3.456
P7 5x50	16537	16612.8	17248.1	84		
				16613.5	2.24×10^{-4}	8.287
				16499.	4.49×10^{-5}	5.15×10^{-1}
				40		
				16613.9	4.49×10^{-4}	16.575
				16499.	9.37×10^{-5}	1.073
				20		
				16616.2	8.99×10^{-4}	33.15
				16499.	1.90×10^{-4}	2.178
				7		
				16616.8	2.24×10^{-5}	82.87
				16499.	6.66×10^{-4}	7.633
ST1 30x60	7772	7839.	8356.5	48		
				7853.4	2.48×10^{-4}	15122.85
				7761.	2.16×10^{-5}	1166.75
				20		
				7855.4	4.96×10^{-4}	30245.69
				7761.	6.91×10^{-5}	3024.23
				13		
				7858.2	7.44×10^{-4}	45368.54
				7741.	1.19×10^{-4}	5945.21
				6		
				7861.4	1.48×10^{-3}	90737.07
				7706.	3.80×10^{-4}	20453.36

Table 1

continue ... 4/5

Tab. 1 - cont.

Prob.	v(P)	PL	$v(SR_w^0)$	k^* [for 4 values of t ($w = w^0$)] / $v(SR_w^{k*})$ / v_h	$\bar{\varrho}_w^0(t)$ / $\bar{\varrho}_w^{k*}(t)$	t/λ_w^0 / t/λ_w^{k*}
ST2 30x60	8722	8773	9123.29	60		
				8775.9	2.10×10^{-4}	16001.44
				8722.	5.17×10^{-6}	694.52
				42		
				8774.8	4.20×10^{-4}	32002.88
				8722.	1.51×10^{-5}	1827.596
				21		
				8774.6	7.00×10^{-4}	53338.14
				8722.	3.19×10^{-5}	4250.105
				6		
				8778.	2.10×10^{-3}	160014.4
				8709.	4.11×10^{-4}	66341.57

Table 1
