

An Exact Polynomial Algorithm for the 0/1 Knapsack Problem

Edouard Petrounevitch, PhD*

*Optimset, 207 Galloway Rd., Unit 401, Toronto ON, M1E 4X3, Canada
(Patent Pending)*

Abstract

A Systematic Decomposition and Multilevel Optimization Search method was developed as an optimization technique to solve the 0/1 Knapsack Problem in polynomial time. This method searches the vast solution space for the focused set of Pareto optimal alternatives. At each level of optimization it employs a forecasting procedure to eliminate the non-productive subsets of items with the weight sum exceeding the knapsack capacity, a Pareto optimization procedure to eliminate non-optimal solutions, and a duplicate elimination procedure to exclude all but one of the subsets of items with the identical weights and values. This optimization approach tremendously reduces the scope of the problem and provides a practitioner with a simple algorithm to carry out an effective search for the exact knapsack loading solutions with time complexity $O((Q / Wu)^2)$, where Wu - the unit of weight of items available for loading (in g, kg, etc), Q - total weight of all items available for loading (in Wu).

Keywords: Knapsack Problem; Exact polynomial algorithm; Combinatorial optimization; Pareto optimization

1. Introduction

The 0/1 Knapsack Problem, also known as the Cargo-Loading Problem, has a wide variety of applications including financial, industrial, computer science, internet security, naval, aerospace, etc. just to name a few.

Consider loading a truck with a subset of items from the initial set of n items available for shipment. Each item has a weight w_j and a value p_j , where $j = 1$ to n . The maximum cargo weight (truck capacity) is W .

The goal is to determine which of the items must be shipped to maximize the value P of the loading without exceeding the maximum weight W of the cargo.

Mathematically the 0/1 Knapsack Problem may be formulated as:

$$\begin{aligned} \text{maximize } P &= \sum_{j=1}^n p_j x_j & \text{subject to } \sum_{j=1}^n w_j x_j &\leq W \\ x_j &\in \{0,1\}; & j &= 1, \dots, n, \end{aligned}$$

where x_j is a binary variable equal to 1 if the item j should be loaded into the knapsack (truck), and 0 otherwise.

The references [1, 2] provide a comprehensive overview of the current methods and techniques available for the solution of the Knapsack Problem.

It is commonly thought that since the Knapsack Problem belongs to the family of NP-hard problems, it is very unlikely that we can ever devise a polynomial-time algorithm for it.

Using a novel approach, we developed a method for obtaining an exact polynomial-time solution for the 0/1 Knapsack Problem.

*E-mail address: optimset@hotmail.com

2. Solution development

The Systematic Decomposition and Multilevel Optimization Search (SDMOS) method was developed to solve the 0/1 Knapsack Problem in polynomial time.

This method searches the vast space of possible solutions for a focused set of Pareto optimal alternatives containing no duplicate elements with identical weights and values.

SDMOS method divides the search into three discrete phases:

1. Multilevel Set Decomposition
2. Multilevel Subset Synthesis
3. Multilevel Optimization, comprised of:
 - 3.1. Forecasting
 - 3.2. Enhanced Pareto Optimization, including:
 - 3.2.1. Pareto Optimization
 - 3.2.2. Duplicate Elimination.

2.1 Multilevel Set Decomposition

The Multilevel Set Decomposition phase uses the Decomposition Table (DT) model, which has the following form:

Table1
Decomposition Table

Level	List 1			List 2		
	Item	Weight	Value	Item	Weight	Value
1	q ₁₁	w ₁₁	p ₁₁	q ₁₀	0	0
2	q ₂₁	w ₂₁	p ₂₁	q ₂₀	0	0

Table 1 (Continued)

Level	List 1			List 2		
	Item	Weight	Value	Item	Weight	Value
...
...
n	q_{n1}	w_{n1}	p_{n1}	q_{n0}	0	0

The number of rows of the DT is equal to n . Each row of the DT contains the number j of the optimization level, one real item q_{j1} from the List1 of n items available for loading into the knapsack, and its corresponding mock item q_{j0} from the List 2 of items having zero weight and value $w_j = p_j = 0$. All n items from the List 1 are sorted in the descending order of weight. None of them has a weight exceeding the knapsack capacity W .

Multilevel Subset Synthesis

In order to load the knapsack, exactly one item (real or mock) from each row of the DT has to be chosen. The Multilevel Subset Synthesis phase starts by combining the items from the first row of the DT with each item from the second row of the DT. Resulting combinations of items will comprise the list of alternative loading solutions S_2 on the second level of the synthesis. If this procedure is repeated until the last row of elements is incorporated (combined) the number of possible alternative solutions may become astronomical, even for a comparatively small-scale knapsack problem. Therefore, Multilevel Optimization phase is used to prevent that from happening.

2.3 Multilevel Optimization

For reducing the number of alternatives the Multilevel Optimization phase can be applied at each level of the synthesis. It employs a Forecasting procedure to eliminate the non-productive subsets having the weight sum exceeding W , a Pareto Optimization procedure to eliminate non-optimal solutions, and a Duplicate Elimination procedure to exclude all but one of the combinations with identical weights and values.

Example1. Let us assume that we have an initial set of 4 items sorted in the descending order of their weights (Table 2, List 1). Our aim is to find a subset of items such that the corresponding value sum is maximized without exceeding the knapsack capacity $W=10$.

Table2

Example1

Level	List 1			List 2		
	Item	Weight	Value	Item	Weight	Value
1	q_{11}	1	20	q_{10}	0	0
2	q_{21}	2	15	q_{20}	0	0
3	q_{31}	3	35	q_{30}	0	0
4	q_{41}	5	30	q_{40}	0	0

To start the Multilevel Subset Synthesis, we need to combine each item from the first row of DT with each item from the second row. Resulting combinations (elements) then comprise the list $S2$ on the level 2 of the synthesis. In round brackets the total weight and value are shown for each corresponding combination.

$$S2 \{q_{11} q_{21} (3,35); q_{11} q_{20} (1,20); q_{10} q_{21} (2,15); q_{10} q_{20} (0,0)\}.$$

This list doesn't contain elements exceeding $W > 10$, no duplicate elements; however it includes both Pareto optimal elements $q_{11} q_{21}$, $q_{11} q_{20}$, $q_{10} q_{20}$, and non-Pareto optimal element $q_{10} q_{21}$ (Fig. 1).

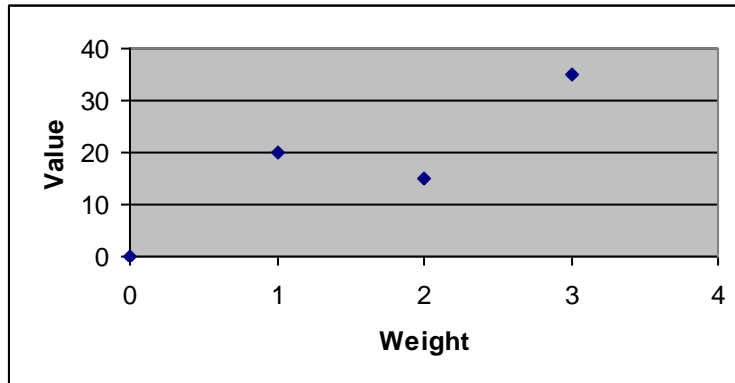


Fig.1. The combinations of items comprising the list $S2$ on the level 2 of the synthesis

Further knapsack loading involves adding the remaining items from the levels 3 and then 4 of the DT to the elements of the list $S2$. Regardless of which item(s) from the levels 3 and 4 will be added to the non-Pareto optimal element $q_{10} q_{21}$, all further combinations involving this element will also be non-Pareto optimal. Thus, because non-Pareto optimal elements eventually produce non-optimal solutions, they should be eliminated from the further knapsack loading process as worthless. As a result, the final list (Fig. 2) of optimal elements on the level 2 of the synthesis is

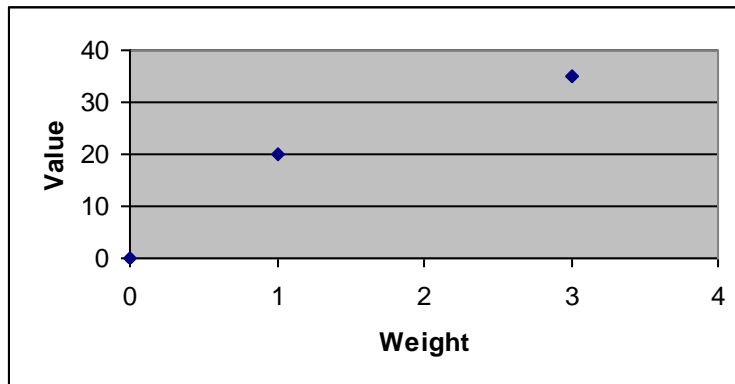


Fig.2. The final list $S2(opt)$ of optimal combinations of items on the level 2 of the synthesis

$S2(opt) \{q_{11} q_{21} (3,35); q_{11} q_{20} (1,20); q_{10} q_{20} (0,0)\}$.

Multilevel Subset Synthesis continues with the next level (3). All possible combinations of elements in the list $S2(opt)$ with each of the items from the third row of the DT generate the list $S3$ on the level 3 of the synthesis

$S3 \{q_{11} q_{21} q_{31} (6,70); q_{11} q_{21} q_{30} (3,35); q_{11} q_{20} q_{31} (4,55); q_{11} q_{20} q_{30} (1,20); q_{10} q_{20} q_{31} (3,35); q_{10} q_{20} q_{30} (0,0)\}$.

The list $S3$ (Fig. 3) contains the duplicate elements $q_{11} q_{21} q_{30} (3,35)$ and $q_{10} q_{20} q_{31} (3,35)$ with identical weights and values.

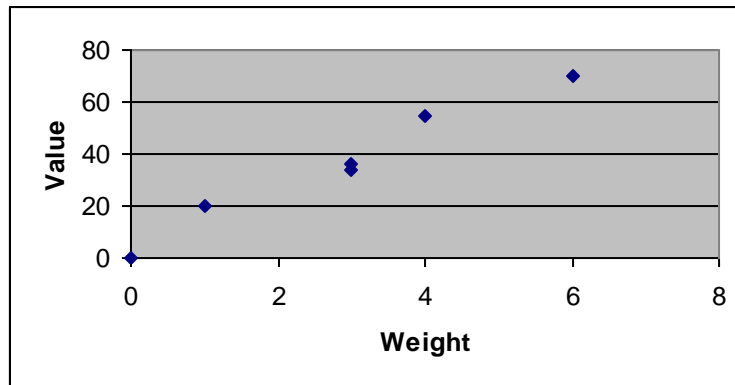


Fig.3. The combinations of items comprising the list $S3$ on the level 3 of the synthesis

As long as we are seeking the single optimal loading, one of these elements should be eliminated. Remaining elements comprise the final list $S3(opt)$ (Fig. 4) of optimal elements on the level 3 of the synthesis is

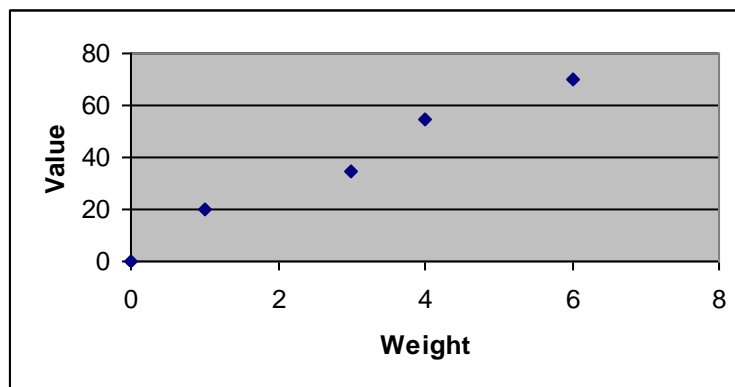


Fig.4. The final list $S3(opt)$ of optimal combinations of items on the level 3 of the synthesis

$S3(opt) \{q_{11} q_{21} q_{31} (6,70); q_{11} q_{21} q_{30} (3,35); q_{11} q_{20} q_{31} (4,55); q_{11} q_{20} q_{30} (1,20); q_{10} q_{20} q_{30} (0,0)\}$.

Note that this list does not contain non-Pareto optimal elements. Finally, all possible combinations of elements from the list $S3(opt)$ with each item from the fourth row of the DT create the list $S4$ (Fig. 5) on the level 4 of the Multilevel Subset Synthesis

$S4 \{ q_{11} q_{21} q_{31} q_{41} (11,100); q_{11} q_{21} q_{31} q_{40} (6,70); q_{11} q_{21} q_{30} q_{41} (8,65); q_{11} q_{21} q_{30} q_{40} (3,35); q_{11} q_{20} q_{31} q_{41} (9,85); q_{11} q_{20} q_{31} q_{40} (4,55); q_{11} q_{20} q_{30} q_{41} (6,50); q_{11} q_{20} q_{30} q_{40} (1,20); q_{10} q_{20} q_{30} q_{41} (5,30); q_{10} q_{20} q_{30} q_{40} (0,0); \}$.

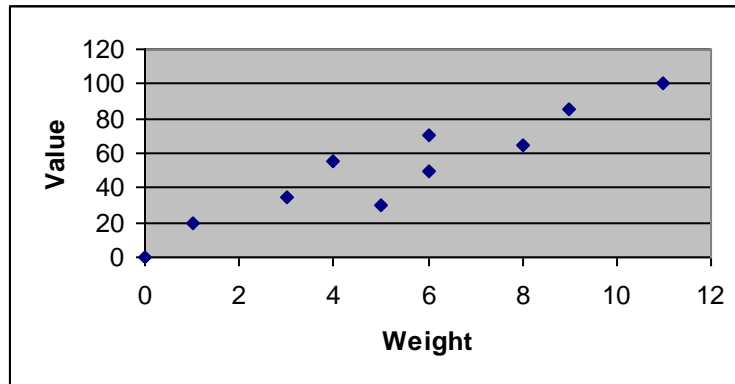


Fig.5. The combinations of items comprising the list $S4$ on the level 4 of the synthesis

Applying the Forecasting procedure to the list $S4$, the element $q_{11} q_{21} q_{31} q_{41}$ should be eliminated because it exceeded the maximum Knapsack capacity; non-Pareto optimal elements $q_{10} q_{20} q_{30} q_{41}$, $q_{11} q_{20} q_{30} q_{41}$, $q_{11} q_{21} q_{30} q_{41}$ should be eliminated as well. Then, the final list $S4(opt)$ of optimal elements on the level 4 of the synthesis (Fig. 6) will be:

$S4(opt) \{ q_{11} q_{21} q_{31} q_{40} (6,70); q_{11} q_{21} q_{30} q_{40} (3,35); q_{11} q_{20} q_{31} q_{41} (9,85); q_{11} q_{20} q_{31} q_{40} (4,55); q_{11} q_{20} q_{30} q_{40} (1,20); q_{10} q_{20} q_{30} q_{40} (0,0); \}$.

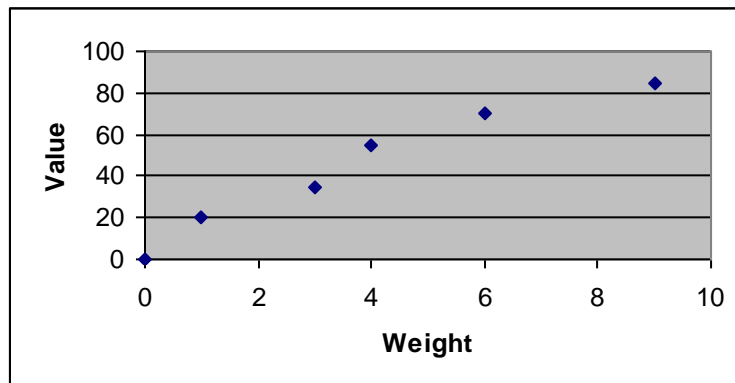


Fig.6. The final list $S4(opt)$ of optimal combinations of items on the level 4 of the synthesis

The sought optimal combination of items for the knapsack loading is $q_{11} q_{20} q_{31} q_{41}$. It has the maximized value sum of 85 without exceeding the capacity of the knapsack $W = 10$. Since the item q_{20} is a mock one, the final optimal subset of items for knapsack loading is

$$S(opt) \{ q_{11} q_{31} q_{41} (9,85) \}.$$

Note. In the preceding loading example the only possible weights of the elements of sets $S2(opt)$ (FIG. 2), $S3(opt)$ (FIG. 4), and $S4(opt)$ (FIG. 6) cannot be other than $0,1,2,3,4,5,6,7,8,9,10$. Consequently, the cardinality of these sets cannot exceed 11.

2.4 The SDMOS method in general

First we shall introduce the following definitions:

- Wu - the unit of weight of items available for loading (in g, kg, etc)
- Q - total weight of all n items available for loading (in Wu).

In general, the list $S1$ of the alternative loading solutions on the level 1 contains the real and mock items from the first row of the DT

$$S1 \{ q_{11} (w_{11}, p_{11}); q_{10} (0,0) \} = R1=2. \quad (1)$$

To find the list Sj of the alternative loading solutions on the level $j = 2, \dots, n$ we need to combine each element from the list $S(j-1)(opt)$ with each of two elements from the list $Rj \{ q_{j1} (w_{j1}, p_{j1}); q_{j0} (0,0) \}$ that contains both real and mock items from the raw j of the DT

$$Sj = S(j-1)(opt) \times Rj. \quad (2)$$

This formula represents the Cartesian product of elements of the list $S(j-1)(opt)$ and elements of the list Rj .

As a result of applying the forecasting and enhanced Pareto optimization procedures, the set $S(j-1)(opt)$ contains only Pareto optimal combinations of items on the level $(j-1)$ with no duplicate elements. These combinations of items will differ from each other by no less than 1 unit of weight Wu .

Let us assume the unique (extreme) case scenario when all n items available for loading have an equal weight Wu . Then, the number of these items and, correspondingly, the number of levels of optimization (synthesis) is n_{max}

$$n_{max} = Q / Wu. \quad (3)$$

On the other hand, the only possible weight of the elements of the set $S(j-1)(opt)$ on any level $j = 2, \dots, n_{max}$ of the synthesis cannot be other than $0, 1, 2, \dots, (n_{max} - 1), n_{max}$. Consequently, the cardinality of these sets cannot exceed $(1 + Q / Wu)$.

Hence, the number of elements in the set S_j (2) on any level $j = 2, \dots, n_{max}$ will not exceed

$$Card(S_j) \leq 2(1+Q/Wu). \quad (4)$$

Thus, in this case the number $C(n_{max})$ of all alternative combinations of items to be evaluated for solving the 0/1 Knapsack Problem is

$$C(n_{max}) \leq R1 + Card(S_j)(n_{max} - 1) \leq 2(Q/Wu)^2. \quad (5)$$

If not all n items available for loading have an equal weight Wu , then the number of these items will be

$$n < n_{max} \quad (6)$$

and, obviously, the number $C(n)$ of all alternative combinations of items to be evaluated for solving the 0/1 Knapsack Problem will not exceed (5).

As a result, for any knapsack capacity W and any number n of items available for loading, the number $C(n)$ of all alternative combinations of items to be evaluated for solving the 0/1 Knapsack Problem is

$$C(n) \leq 2(Q/Wu)^2. \quad (7)$$

Example2. Let us suppose that we have an initial set of 1000 items with a total weight of 2000kg. Our goal is to find a subset of items such that the corresponding value sum is maximized without exceeding the knapsack capacity $W=1900kg$.

A straightforward way to solve this problem by an exhaustive search involves creation and evaluation of about 2^{1000} alternatives. Our approach requires creation and evaluation of less than $2(2000)^2$ or $\approx 10^7$ alternative combinations.

Example3. Let us suppose that we have an initial set of 1800 items with a total weight of 2000kg. Our goal is to find a subset of items such that the corresponding value sum is maximized without exceeding the knapsack capacity $W=1900kg$.

A straightforward way to solve this problem by an exhaustive search involves creation and evaluation of about 2^{1800} alternatives. Our approach requires creation and evaluation of less than $2(2000)^2$ or $\approx 10^7$ alternative combinations.

An exact polynomial algorithm for solving the 0/1 Knapsack Problem

An exact polynomial algorithm for solving the 0/1 Knapsack Problem is as follows:

1. generate an initial list of n items available for knapsack loading wherein a weight of each item must not exceed the knapsack capacity W and all n items are sorted in the descending order of their weights

2. create a Decomposition Table (DT) with the initial list of n items available for knapsack loading in such a way that
 - 2.1. a number of rows of the DT is equal to n
 - 2.2. each row of the DT contains the number of the optimization level j from 1 to n , one real item k_{j1} from the list of n items available for loading into the knapsack, and its corresponding mock item k_{j0} from a list of mock items all having zero weight and value
3. $j := 1$
4. generate the list R_j containing both items from the row j of the DT

$$R_j \{k_{j1} (w_{j1}, p_{j1}); k_{j0} (0,0)\}$$

5. $S_j(opt) := R_j$
6. if $n < 2$
7. delete mock item k_{j0} from $S_j(opt)$
8. $S(opt) := S_j(opt)$
9. print $S(opt)$ as the sought optimal combination of items for the knapsack loading
10. else
11. for $j := 2$ to n
12. find the list S_j of alternative loading solutions on the level j by

$$S_j = S_{(j-1)}(opt) \times R_j$$

13. delete from S_j the combinations of items having the weight sum exceeding the knapsack capacity W
14. find the subset of Pareto optimal combinations of items $S_j(par)$ from the list S_j
15. delete from $S_j(par)$ all but one of the combinations with identical weights and values
16. $S_j(opt) := S_j(par)$
17. $j = j + 1$
18. end for
19. find the combination of items with the maximum value $Sp(max)$ from the list $S_j(opt)$
20. delete mock items from $Sp(max)$
21. $S(opt) := Sp(max)$
22. print $S(opt)$ as the sought optimal combination of items for the knapsack loading
23. end if.

4. Computational results

The described algorithm was implemented in Java on Windows platform using a Pentium-IV 1.8 GHz CPU with a memory of 512 MB. For the demonstration purposes we used the initial set of 740 items (Table 3). We found two subsets of optimal cargoes ranged in weights:

- $(W1) = 6000$ (Table 4)
- $(W2) = 12000$ (Table 5)

Table 3
Initial set of items for cargo optimization (740 items)

Item	Weight	Value	Item	Weight	Value
A41	1	300000	G54	29	1900000
A42	1	350000	G56	30	840000
A43	1	350000	G59	30	1165000
A44	1	365000	G60	30	1300000
A45	1	400000	G62	31	375000
A46	1	409000	G65	31	600000
A47	1	425000	G70	31	1300000
A48	1	450000	G72	32	632500
A49	1	450000	G74	32	907500
A50	1	467500	G75	32	1025000
A51	1	700000	G76	32	6000000
A52	1	700000	G78	33	715000
A53	1	2523598	G80	34	500000
A54	2	233333	G90	35	675000
A55	2	300000	G95	36	1200000
A56	2	300000	G96	36	2000000
A57	2	300000	G97	36	2100000
A58	2	385000	G98	37	600000
A59	2	400000	H10	39	1320000
A60	2	400000	H12	39	1750000
A61	2	400000	H14	40	750000
A62	2	437500	H18	40	3500000
A63	2	550000	H21	42	350000
A64	2	550000	H24	42	750000
A65	2	580000	H26	43	850000
A66	2	775000	H28	43	4300000
A67	2	875000	H34	45	925000
A68	2	925000	H40	46	800000
A69	2	925000	H41	46	800000
A70	2	925000	H43	46	1000000
A71	2	975000	H53	48	516667
A72	2	1025000	H57	49	1320000
A73	2	1800000	H59	49	2400000
A74	3	200000	H60	50	425000
A75	3	325000	H69	51	1300000
A76	3	350000	H74	53	2500000
A77	3	366667	H75	54	675000
A78	3	400000	H79	56	400000
A79	3	400000	H92	61	2500000
A80	3	415000	H99	63	2500000
A81	3	415000	J03	65	2250000
A82	3	425000	J09	67	2250000
A83	3	430000	J14	70	1350000

Table 3(Continued)

Item	Weight	Value	Item	Weight	Value
A84	3	475000	J16	71	1350000
A85	3	516667	J23	74	3500000
A86	3	650000	J26	77	3969123
A87	3	850000	J27	78	4000000
A88	3	850000	J35	86	10000000
A89	3	1000000	D60	1	290000
A90	3	1800000	D61	1	300000
A91	4	266667	D63	1	330000
A92	4	300000	D68	1	375000
A93	4	350000	D70	1	385000
A94	4	375000	D71	1	400000
A95	4	450000	D72	1	400000
A96	4	450000	D75	1	410000
A97	4	523687	D86	1	675000
A98	4	525000	D87	1	850000
A99	4	625000	D89	1	1025000
B00	4	650000	D90	2	283333
B01	4	725000	D91	2	325000
B02	4	1300000	D96	2	350000
B03	4	1650000	D99	2	375000
B04	4	2000000	E00	2	375000
B05	5	400000	E03	2	550000
B06	5	400000	E04	2	650000
B07	5	402500	E05	2	750000
B08	5	475000	E06	2	800000
B09	5	525000	E07	2	825000
B10	5	600000	E08	2	875000
B11	5	605000	E10	2	1400000
B12	5	675000	E15	3	375000
B13	5	990000	E17	3	425000
B14	5	2750000	E18	3	500000
B15	6	220000	E22	3	625000
B16	6	250000	E23	3	687500
B17	6	300000	E25	3	1650000
B18	6	330000	E26	4	277083
B19	6	345000	E37	4	750000
B20	6	425000	E48	5	450000
B21	6	425000	E51	5	847000
B22	6	450000	E52	6	325000
B23	6	700000	E53	6	400000
B24	6	1210000	E55	6	425000
B25	6	1400000	E56	6	475000
B26	6	1700000	E57	6	775000
B27	6	1731137	E59	6	800000

Table 3(Continued)

Item	Weight	Value	Item	Weight	Value
B28	7	260000	E61	6	1025000
B29	7	425000	E63	6	1350000
B30	7	475000	E73	7	875000
B31	7	550000	E74	8	475000
B32	7	750000	E75	8	475000
B33	7	800000	E78	8	750000
B34	7	850000	E81	8	1100000
B35	7	1066000	E87	9	425000
B36	7	1200000	E90	9	660000
B37	7	1400000	E91	9	775000
B38	7	2500000	E92	9	800000
B39	8	426667	E97	10	715000
B40	8	450000	E99	11	350000
B41	8	500000	F00	11	400000
B42	8	550000	F04	11	700000
B43	8	650000	F06	11	825000
B44	8	825000	F07	11	850000
B45	8	875000	F11	12	440000
B46	8	950000	F15	12	525000
B47	8	975000	F18	12	750000
B48	8	1000000	F20	12	875000
B49	8	1200000	F22	13	250000
B50	9	400000	F23	13	450000
B51	9	425000	F26	13	467500
B52	9	440000	F28	13	550000
B53	9	450000	F33	14	475000
B54	9	485000	F35	14	875000
B55	9	600000	F37	14	950000
B56	9	625000	F39	15	400000
B57	9	1000000	F47	16	435000
B58	9	1075000	F51	16	800000
B59	9	1250000	F54	17	325000
B60	9	1500000	F56	17	400000
B61	10	216667	F57	17	400000
B62	10	425000	F58	17	440000
B63	10	600000	F60	17	522500
B64	10	650000	F62	17	3500000
B65	10	775000	F65	18	425000
B66	10	850000	F66	18	440000
B67	10	1075000	F68	18	495000
B68	10	1100000	F69	18	800000
B69	10	1650000	F73	19	510000
B70	11	360000	F78	20	430000
B71	11	450000	F79	20	475000

Table 3(Continued)

Item	Weight	Value	Item	Weight	Value
B72	11	475000	F85	20	1200000
B73	11	475000	F88	21	587500
B74	11	484000	F90	21	1250000
B75	11	800000	F92	21	3300000
B76	11	875000	F93	22	266667
B77	11	875000	F94	22	350000
B78	11	1350000	F95	22	350000
B79	11	1800000	F96	22	435000
B80	12	293333	F97	22	650000
B81	12	500000	F99	22	750000
B82	12	675000	G00	22	800000
B83	12	700000	G03	22	3500000
B84	12	975000	G04	23	350000
B85	12	2000000	G05	23	350000
B86	12	2600000	G07	23	500000
B87	13	600000	G11	23	900000
B88	13	700000	G12	24	575000
B89	13	1500000	G14	24	875000
B90	13	2500000	G18	25	400000
B91	14	400000	G21	25	683333
B92	14	400000	G23	25	700000
B93	14	500000	G25	25	1025000
B94	14	630000	G32	26	1450000
B95	14	1000000	G36	27	875000
B96	14	1100000	G37	27	875000
B97	14	1200000	G39	27	1100000
B98	14	1200000	G40	28	500000
B99	15	900000	G44	28	975000
C00	15	1400000	G48	28	2600000
C01	16	350000	G49	29	450000
C02	16	400000	G51	29	600000
C03	16	700000	G52	29	685000
C04	16	762500	G55	30	575000
C05	16	900000	G57	30	975000
C06	16	925000	G58	30	1000000
C07	16	975000	G61	30	3000000
C08	16	3950000	G64	31	600000
C09	17	350000	G67	31	800000
C10	17	450000	G68	31	825000
C11	17	632500	G79	33	2300000
C12	17	660000	G81	34	592500
C13	17	1233333	G82	34	600000
C14	17	1325000	G84	34	1025000
C15	17	1800000	G86	34	2600000

Table 3(Continued)

Item	Weight	Value	Item	Weight	Value
C16	18	650000	G87	35	500000
C17	18	925000	H01	37	1700000
C18	18	1600000	H03	38	425000
C19	18	2500000	H06	39	330000
C20	18	3100000	H07	39	400000
C21	19	325000	H09	39	1150000
C22	19	875000	H15	40	925000
C23	19	1480000	H17	40	1025000
C24	20	575000	H20	41	1650000
C25	20	675000	H22	42	500000
C26	20	797500	H30	44	890000
C27	20	800000	H31	44	1000000
C28	20	1967290	H32	44	1550000
C29	20	2000000	H35	45	1050000
C30	21	450000	H37	45	1500000
C31	21	700000	H39	46	500000
C32	22	450000	H42	46	875000
C33	22	2300000	H44	46	1800000
C34	22	2750000	H46	46	2500000
C35	23	233333	H48	47	850000
C36	23	3100000	H50	47	1700000
C37	24	1500000	H54	48	975000
C38	24	1650000	H58	49	1500000
C39	24	1800000	H61	50	600000
C40	24	3800000	H62	50	700000
C41	25	266667	H65	51	750000
C42	25	1400000	H67	51	975000
C43	25	1500000	H68	51	1233333
C44	25	1575000	H70	51	9000000
C45	25	1700000	H71	52	800000
C46	25	1850000	H72	53	350000
C47	26	340000	H73	53	750000
C48	26	1900000	H76	54	852500
C49	26	6680000	H77	54	6000000
C50	27	1000000	H78	55	600000
C51	27	1600000	H81	56	1700000
C52	28	742000	H82	56	1800000
C53	28	750000	H83	57	2050000
C54	29	1900000	H84	58	1750000
C55	29	4152579	H85	59	2000000
C56	30	1400000	H88	59	8500000
C57	30	1500000	H89	60	800000
C58	30	1804000	H90	60	925000
C59	30	2750000	H91	60	3000000

Table 3 (Continued)

Item	Weight	Value	Item	Weight	Value
C60	31	400000	H94	62	1750000
C61	31	1100000	H95	62	1850000
C62	32	450000	H96	62	2000000
C63	32	800000	H97	63	605000
C64	32	1800000	J01	65	825000
C65	33	600000	J02	65	925000
C66	33	950000	J05	66	5000000
C67	33	1500000	J07	67	640000
C68	34	1000000	J08	67	975000
C69	34	2400000	J10	68	625000
C70	34	3000000	J11	68	3500000
C71	34	3735849	J12	70	750000
C72	35	825000	J15	70	3000000
C73	35	1500000	J17	71	3136825
C74	35	2000000	J19	72	4300000
C75	36	500000	J20	73	5000000
C76	36	975000	J21	73	6000000
C77	36	1625000	J22	73	7000000
C78	37	3100000	J28	78	4200000
C79	38	875000	J29	79	6000000
C80	38	3250000	J30	81	2000000
C81	39	4763802	J31	81	6000000
C82	40	240000	D66	1	350000
C83	40	2500000	D67	1	350000
C84	40	3000000	D73	1	400000
C85	40	3400000	D74	1	400000
C86	41	1000000	D76	1	425000
C87	42	2250000	D78	1	450000
C88	42	2425000	D79	1	450000
C89	44	750000	D80	1	450000
C90	45	2250000	D82	1	500000
C91	49	3000000	D85	1	650000
C92	51	1700000	E02	2	500000
C93	52	4000000	E12	3	275000
C94	52	6000000	E14	3	350000
C95	54	1800000	E20	3	525000
C96	55	2500000	E21	3	600000
C97	55	3350000	E24	3	825000
C98	57	5267500	E28	4	300000
C99	62	3425000	E29	4	325000
D00	72	900000	E31	4	375000
D01	73	6000000	E36	4	550000
D62	1	325000	E38	4	800000
D64	1	350000	E43	5	200000

Table 3 (Continued)

Item	Weight	Value	Item	Weight	Value
D65	1	350000	E44	5	350000
D69	1	375000	E46	5	375000
D77	1	450000	E58	6	775000
D81	1	500000	E72	7	850000
D83	1	585000	E77	8	660000
D84	1	630000	E80	8	850000
D88	1	875000	E82	8	1500000
D92	2	325000	E85	9	350000
D93	2	333333	E93	9	1050000
D94	2	333333	E96	10	375000
D95	2	345000	F05	11	800000
D97	2	350000	F09	12	345000
D98	2	350000	F12	12	450000
E01	2	450000	F13	12	500000
E09	2	878697	F14	12	514250
E11	3	200000	F16	12	550000
E13	3	312500	F21	12	1650000
E16	3	420000	F27	13	535000
E19	3	500000	F32	14	350000
E27	4	300000	F38	14	1557651
E30	4	335000	F45	16	240000
E32	4	390000	F55	17	357500
E33	4	400000	F59	17	500000
E34	4	450000	F63	18	276667
E35	4	525000	F64	18	400000
E39	4	3500000	F70	18	875000
E40	5	150000	F71	18	950000
E41	5	150000	F74	19	525000
E42	5	183333	F76	19	850000
E45	5	352000	F80	20	700000
E47	5	425000	F82	20	800000
E49	5	725000	F83	20	936916
E50	5	825000	F84	20	1100000
E54	6	400000	F87	21	400000
E60	6	957755	F89	21	850000
E62	6	1100000	G02	22	1600000
E64	7	233333	G06	23	450000
E65	7	300000	G08	23	550000
E66	7	400000	G15	24	1000000
E67	7	450000	G17	25	350000
E68	7	450000	G22	25	690000
E69	7	475000	G24	25	850000
E70	7	625000	G26	25	1500000
E71	7	775000	G27	26	400000

Table 3 (Continued)

Item	Weight	Value	Item	Weight	Value
E76	8	535000	G28	26	525000
E79	8	850000	G29	26	600000
E83	8	3204011	G38	27	1000000
E84	9	290000	G42	28	600000
E86	9	400000	G63	31	550000
E88	9	450000	G66	31	675000
E89	9	625000	G69	31	850000
E94	9	1650000	G71	32	450000
E95	10	350000	G73	32	800000
E98	11	250000	G77	33	200000
F01	11	425000	G83	34	750000
F02	11	425000	G85	34	1925000
F03	11	465000	G88	35	650000
F08	12	233333	G89	35	650000
F10	12	400000	G91	35	800000
F17	12	700000	G92	35	1700000
F19	12	750000	G93	35	1800000
F24	13	450000	G94	35	2150000
F25	13	450000	G99	37	600000
F29	13	550000	H00	37	850000
F30	13	750000	H02	37	2161634
F31	13	1600000	H04	38	850000
F34	14	825000	H05	38	3870362
F36	14	875000	H08	39	400000
F40	15	400000	H11	39	1700000
F41	15	600000	H13	39	1842852
F42	15	725000	H16	40	1000000
F43	15	750000	H19	41	1050000
F44	15	1800000	H23	42	660000
F46	16	375000	H25	42	2400000
F48	16	625000	H27	43	2200000
F49	16	700000	H29	44	875000
F50	16	725000	H33	44	4600000
F52	16	975000	H36	45	1100000
F53	16	1550000	H38	45	1800000
F61	17	650000	H45	46	2400000
F67	18	475000	H47	46	2600000
F72	19	400000	H49	47	850000
F75	19	800000	H51	47	1700000
F77	19	1200000	H52	47	2466823
F81	20	750000	H55	48	1000000
F86	20	1200000	H56	48	1800000
F91	21	2100000	H63	50	1000000
F98	22	650000	H64	50	1100000

Table 3 (Continued)

Item	Weight	Value	Item	Weight	Value
G01	22	940000	H66	51	800000
G09	23	825000	H80	56	700000
G10	23	850000	H86	59	2600000
G13	24	610918	H87	59	5500000
G16	24	1100000	H93	62	1250000
G19	25	500000	H98	63	1500000
G20	25	525000	J00	64	8500000
G30	26	650000	J04	66	2100000
G31	26	800000	J06	66	5000000
G33	26	1500000	J13	70	1000000
G34	27	400000	J18	72	625000
G35	27	750000	J24	75	1100000
G41	28	516667	J25	75	1450000
G43	28	800000	J32	84	2750000
G45	28	1100000	J33	84	2900000
G46	28	1300000	J34	85	5450000
G47	28	2250000	J36	91	5000000
G50	29	525000	J37	94	8000000
G53	29	1100000	J38	96	10359852

Table 4

The optimal cargo in range of $WI = 6000$

A41/A42/A43/A44/A45/A46/A47/A48/A49/A50/A51/A52/A53/A54/A55/A56/A57/A58/ A59/A60/A61/A62/A63/A64/A65/A66/A67/A68/A69/A70/A71/A72/A73/A74/A75/A76/ A77/A78/A79/A80/A81/A82/A83/A84/A85/A86/A87/A88/A89/A90/A91/A92/A93/A94/ A95/A96/A97/A98/A99/B00/B01/B02/B03/B04/B05/B06/B07/B08/B09/B10/B11/B12/ B13/B14/B17/B18/B19/B20/B21/B22/B23/B24/B25/B26/B27/B29/B30/B31/B32/B33/ B34/B35/B36/B37/B38/B39/B40/B41/B42/B43/B44/B45/B46/B47/B48/B49/B54/B55/ B56/B57/B58/B59/B60/B63/B64/B65/B66/B67/B68/B69/B75/B76/B77/B78/B79/B82/ B83/B84/B85/B86/B88/B89/B90/B95/B96/B97/B98/B99/C00/C05/C06/C07/C08/C13/ C14/C15/C17/C18/C19/C20/C23/C28/C29/C33/C34/C36/C37/C38/C39/C40/C42/C43/ C44/C45/C46/C48/C49/C51/C54/C55/C58/C59/C64/C69/C70/C71/C74/C78/C80/C81/ C83/C84/C85/C87/C88/C91/C93/C94/C97/C98/C99/D01/D60/D61/D62/D63/D64/D65/ D66/D67/D68/D69/D70/D71/D72/D73/D74/D75/D76/D77/D78/D79/D80/D81/D82/D83/ D84/D85/D86/D87/D88/D89/D90/D91/D92/D93/D94/D95/D96/D97/D98/D99/E00/E01/ E02/E03/E04/E05/E06/E07/E08/E09/E10/E11/E12/E13/E14/E15/E16/E17/E18/E19/ E20/E21/E22/E23/E24/E25/E26/E27/E28/E29/E30/E31/E32/E33/E34/E35/E36/E37/ E38/E39/E44/E45/E46/E47/E48/E49/E50/E51/E52/E53/E54/E55/E56/E57/E58/E59/ E60/E61/E62/E63/E66/E67/E68/E69/E70/E71/E72/E73/E74/E75/E76/E77/E78/E79/ E80/E81/E82/E83/E89/E90/E91/E92/E93/E94/E97/F04/F05/F06/F07/F17/F18/F19/ F20/F21/F30/F31/F34/F35/F36/F37/F38/F44/F52/F53/F62/F71/F77/F84/F85/F86/ F90/F91/F92/G02/G03/G26/G32/G33/G47/G48/G54/G61/G76/G79/G85/G86/G93/G94/ G96/G97/H02/H05/H18/H25/H27/H28/H33/H45/H46/H47/H52/H70/H77/H87/H88/H91/ J00/J05/J06/J11/J19/J20/J21/J22/J26/J27/J28/J29/J31/J34/J35/J36/J37/J38/
--

Weight = 6000.00

Value = 578627266.00

Time = 60sec

Table 5
The optimal cargo in range of $WI = 12000$

A41/A42/A43/A44/A45/A46/A47/A48/A49/A50/A51/A52/A53/A54/A55/A56/A57/A58/
A59/A60/A61/A62/A63/A64/A65/A66/A67/A68/A69/A70/A71/A72/A73/A74/A75/A76/
A77/A78/A79/A80/A81/A82/A83/A84/A85/A86/A87/A88/A89/A90/A91/A92/A93/A94/
A95/A96/A97/A98/A99/B00/B01/B02/B03/B04/B05/B06/B07/B08/B09/B10/B11/B12/
B13/B14/B15/B16/B17/B18/B19/B20/B21/B22/B23/B24/B25/B26/B27/B28/B29/B30/
B31/B32/B33/B34/B35/B36/B37/B38/B39/B40/B41/B42/B43/B44/B45/B46/B47/B48/
B49/B50/B51/B52/B53/B54/B55/B56/B57/B58/B59/B60/B62/B63/B64/B65/B66/B67/
B68/B69/B70/B71/B72/B73/B74/B75/B76/B77/B78/B79/B80/B81/B82/B83/B84/B85/
B86/B87/B88/B89/B90/B91/B92/B93/B94/B95/B96/B97/B98/B99/C00/C02/C03/C04/
C05/C06/C07/C08/C10/C11/C12/C13/C14/C15/C16/C17/C18/C19/C20/C22/C23/C24/
C25/C26/C27/C28/C29/C31/C33/C34/C36/C37/C38/C39/C40/C42/C43/C44/C45/C46/
C48/C49/C50/C51/C52/C53/C54/C55/C56/C57/C58/C59/C61/C63/C64/C66/C67/C68/
C69/C70/C71/C72/C73/C74/C76/C77/C78/C80/C81/C83/C84/C85/C86/C87/C88/C90/
C91/C92/C93/C94/C95/C96/C97/C98/C99/D01/D60/D61/D62/D63/D64/D65/D66/D67/
D68/D69/D70/D71/D72/D73/D74/D75/D76/D77/D78/D79/D80/D81/D82/D83/D84/D85/
D86/D87/D88/D89/D90/D91/D92/D93/D94/D95/D96/D97/D98/D99/E00/E01/E02/E03/
E04/E05/E06/E07/E08/E09/E10/E11/E12/E13/E14/E15/E16/E17/E18/E19/E20/E21/
E22/E23/E24/E25/E26/E27/E28/E29/E30/E31/E32/E33/E34/E35/E36/E37/E38/E39/
E40/E41/E42/E43/E44/E45/E46/E47/E48/E49/E50/E51/E52/E53/E54/E55/E56/E57/
E58/E59/E60/E61/E62/E63/E64/E65/E66/E67/E68/E69/E70/E71/E72/E73/E74/E75/
E76/E77/E78/E79/E80/E81/E82/E83/E84/E85/E86/E87/E88/E89/E90/E91/E92/E93/
E94/E95/E96/E97/E98/E99/F00/F01/F02/F03/F04/F05/F06/F07/F09/F10/F11/F12/
F13/F14/F15/F16/F17/F18/F19/F20/F21/F23/F24/F25/F26/F27/F28/F29/F30/F31/
F32/F33/F34/F35/F36/F37/F38/F39/F40/F41/F42/F43/F44/F46/F47/F48/F49/F50/
F51/F52/F53/F56/F57/F58/F59/F60/F61/F62/F65/F66/F67/F68/F69/F70/F71/F73/
F74/F75/F76/F77/F79/F80/F81/F82/F83/F84/F85/F86/F88/F89/F90/F91/F92/F97/
F98/F99/G00/G01/G02/G03/G08/G09/G10/G11/G12/G13/G14/G15/G16/G21/G22/G23/
G24/G25/G26/G29/G30/G31/G32/G33/G35/G36/G37/G38/G39/G43/G44/G45/G46/G47/
G48/G52/G53/G54/G56/G57/G58/G59/G60/G61/G67/G68/G69/G70/G73/G74/G75/G76/
G79/G84/G85/G86/G92/G93/G94/G95/G96/G97/H01/H02/H05/H09/H10/H11/H12/H13/
H15/H16/H17/H18/H19/H20/H25/H27/H28/H31/H32/H33/H35/H36/H37/H38/H44/H45/
H46/H47/H50/H51/H52/H56/H57/H58/H59/H68/H69/H70/H74/H77/H81/H82/H83/H84/
H85/H86/H87/H88/H91/H92/H94/H95/H96/H98/H99/J00/J03/J04/J05/J06/J09/J11/
J15/J17/J19/J20/J21/J22/J23/J26/J27/J28/J29/J30/J31/J32/J33/J34/J35/J36/
J37/J38/

Weight = 12000.00 Value = 787174192.00 Time = 120sec

5. Conclusion

The proposed Systematic Decomposition and Multilevel Optimization Search method has been developed as an optimization technique to solve the 0/1 Knapsack Problem in polynomial time. According to this method, for any knapsack capacity W and any number n of items available for loading, the number $C(n)$ of all alternative combinations of items to be evaluated to solve the 0/1 Knapsack Problem is

$$C(n) \leq 2(Q/Wu)^2,$$

where Wu - the unit of weight of all n items available for loading (in g, kg, etc), Q - total weight of all n items (in Wu). Hence, the developed algorithm allows carrying out a search for the exact knapsack loading solutions with time complexity $O((Q / Wu)^2)$.

Acknowledgements

Thanks to Stan Nevski and Vitali Petrounevitch for extensive computational testing of the proposed algorithm and for their very helpful comments and suggestions.

References

- [1] Martello Silvano, Toth Paolo (1990). *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons. ISBN 0-471- 92420-2.
- [2]Hans Kellerer, Ulrich Pferschy, and David Pisinger (2005). *Knapsack Problems*. Springer Verlag. ISBN 3-540-40286-1.