This approach is presented by Djangir Babayev, Fred Glover, Jennifer Ryan. According to this a new and highly efficient algorithm for the Integer knapsack problem is based on a special strategy for aggregating Integer–valued equations. In this, Integer knapsack problem is transformed into an equivalent problem of determining the consistency of a aggregated equation for a parameterized right hand side. Then this is solved by a newly developed algorithm with complexity $O(\min(n\alpha, n+\alpha^2_{1}))$, where $n$ is the no. of variables and $\alpha_{i}$is the smallest coefficient in the aggregated equation.

This approach is better and fast for the harder and several orders problems. Consider The Integer Knapsack Problem

Maximize $\sum_{j \in N} c_{j}x_{j}$ ..........(1) subject to: $\sum_{j \in N} a_{j}x_{j} \leq b$ ..........(2) $x_{j} \geq 0$, integer, $j \in N \equiv (1,2,3,4,..n)$

where $a_{j}, b > 0$, $c_{j} \geq 0$, integer, $j \in N$.

Inequality (2) can be reduced to an equivalent equality by adding slack variable and making the problem (KNP) equivalent to the following Maximum Consistency Problem (MCP)

Determine

Max $c \equiv c^{*}$..............(4)

For which the system of two equations

$\sum_{j \in N} c_{j}x_{j} = c$......(5) $\sum_{j \in N} a_{j}x_{j} = b$......(6)

is consistent under non-negativity and integrality requirements.

The system (5)-(6) can be reduced to one equivalent equation by using Integer Equivalent Aggregation, the principle result of which is formulated as:

Proposition: For an arbitrary system of linear algebraic equations with integer coefficients and right hand sides, and a bounded set of integer non negative solutions, an infinite no. of pairs of integer weights exists, so that any equation, create as a linear combination of the original equations with these weights, has the same set of nonnegative integer solutions as the original systems.

Now the MCP can be changed into an equivalent problem that involves a single equation:

Determine

Max $c \equiv c^{*}$

For which equation $\sum_{j \in N} j_{j}x_{j} = _{-}$..............(7)

is consistent in nonnegative integer solutions, where $_{-}$ and $_{-}$ are nonnegative integers, and $_{-}$ depends on c.

Maximum Consistency Algorithm

Step 1. (Initialization). Determine $_{-}$an upper bound for $c^{*}$, $\geq c^{*}$

Step 2. $c = _{-}$

Step 3. Compute $_{-}$and test consistency of Eq. 7

Step 4. If eq.7 is inconsistent, then $c = c-1$ and return to step 3.

Step 5. If eq.7 is consistent then $c^{*} = c$ and the corresponding solution of eq.7 is a solution to the original knapsack probk (1)-(3). End.

Now we have to define the two Multipliers which are required to aggregate the systems of Eq. 5 &6 into one equivalent equation. They are v (value) for Eq.5 and w (weight) for eq.6. The maximum consistency Algorithm requires the consistency problem to be solved for various values of c, starting from $_{-}$an upper bound for $c^{*}$

Moreover, it is important to generate these multipliers to yield coefficients of the aggregated equation that are positive and whose size is restricted in a manner subsequently specified. For this we have a Theorem which is

We have: $\sum_{j \in N} j_{j}x_{j} = _{-}........(10)$ where $_{-} = vc_{j} + wa_{j}$, $_{N}$, $_{-} = vc + wb........(11)$

Constraints in 11 possesses the same set of nonnegative integer solutions as system 5-6, if v and w are relatively prime integers.
After the calculations, according to this theorem (not shown here), we can summarize that multipliers must satisfy these conditions.

(1). \( v < 0 \)  
(2) \( w > 0 \)  
(3). \( v \) and \( w \) are relatively prime  
(4). \( w + vR > Rb-c \)  
Thus in sum the multipliers \( v \) and \( w \) are defined by the following:

**Procedure M1.**

Select \( v \) as the negative integer of the smallest absolute value satisfying these eq.

(1) \(-v > \left(\frac{\_}{k}\right) - 1\)  
(2) \(-v > \left(\frac{bR-c-1}{k-R}\right)\) with \( k \) given by

\[
k = \begin{cases} 
  \lfloor R \rfloor & \text{if } R < \lfloor R \rfloor \\
  R + 1 & \text{if } R = \lfloor R \rfloor
\end{cases}
\]

Having defined the maximum consistency problem (4)-(6), (3) is reduced to the following Maximum consistency Problem for a single equation

**Determine:** \( \text{max } c - c^* \) for which equation 10 is consistent in nonnegative integer solutions.

Solving the Consistency Problem: At this point The Integer Knapsack Problem has been reduced to the problem MCP(\(_\))\text{ a single equation, which can be solved by Maximum Consistency Algorithm. Now the last algorithm includes solution of the following Consistency Problem CP(\(_\))

Given \( \_ > 0, j \in \mathbb{N} \) Determine : Do there exist nonnegative integers \( x_j \) satisfying equation(10) for a given \( \_ \). Without loss of generality we can assume that the greatest common divisor \( d \) of \( \_1, \_2, \ldots, \_n \) is equal to1. Define \( \_i = \min_{j \in \mathbb{N}} \_j \) and introduce a Boolean function defined for integer nonnegative values of the argument \( r \),

\[
f(r) = \begin{cases} 
  1 & \text{if CP(r) is feasible} \\
  0 & \text{otherwise}
\end{cases}
\]

To solve this we will make use of an approach that allows us to exploit the structure of the monoid \( \{r \in \mathbb{Z} / f(r) = 1\} \) where \( \mathbb{Z} \) denotes the set of non negative integers . For all \( i, 0 \leq i \leq \_1 - 1 \), define

\[
R_i = \min ((r_i \mod \_1) \text{ and } (f(r_i) = 1)), 0 \leq i \leq \_1 - 1
\]

i.e. \( r_i \) is the minimal value of right hand side of eq.(10), which satisfies the condition \( \_ = \_i \mod \_1 \) and for which eq.10 is consistent in nonnegative integer. The role of \( r_i \) in the considered problem is defined by the theorem which states CP(\(_\)) is consistent if and only if \( \_ \geq r_i \mod \_1 \).

After each \( r_i, i = 1,2,3,\ldots,\_1 - 1 \) has been computed and stored the above theorem enable us to check the consistency of eq. for an arbitrary right hand side \( \_ \) simply by comparing the magnitude of \( \_ \) with \( r_i \mod \_1 \).

Consequently to compute \( r_i \) problem GP (i) must be solved for all \( i, 1 \leq i \leq \_1 - 1 \). Now we can solve problem.

**GP (i) by this approach.**

Set up a graph and solve the problem of finding the shortest paths from one specific vertex to all other vertices of this graph. To this end define \( j_i = \arg \min(\_j \mid \_j \mod \_i) \), \( i = 1,2,3,\ldots,\_1 - 1 \). If there is no \( j \) with \( j \mod \_i \), then \( j_i = \infty \). Let \( S = (\_j, \infty) \).

Create a graph with vertex set \( V \) and arc set \( A \) as follows:

\[
V = (0,1,2,3,4,5,\ldots,\_1 - 1) \text{ and } \quad A = (v, w) \mid \text{there exist } v, w \in V \text{ with } v + w \mod \_1 \text{ and } v + w \mod \_1 \text{ is in } S
\]

If \((v, w) \in A\) with \( v \mod \_1 \) satisfying \( v + w \mod \_1 \), then the length of arc \((v, w)\) is \( -\_j \).

The arc set \( A \) of graph \( G \) has a property which can be exploited to decrease the effort to create this graph.

**Conclusion:**

The consistency Approach for solving integer knapsack problem based on Integer Equivalent Aggregation, reduces a system of linear equations with integer Coefficients to a single equation with the same set of nonnegative integer solutions. Thus the Integer knapsack problem is essentially reduced to testing the solvability of a single, linear Diophantine equation.