Probability for the Existence of the Solutions for Constant Sum Integer Programming

Silviu Bârză
Faculty of Mathematics and Informatics
Spiru Haret University
Bucharest, Romania
silviu.barza@gmail.com

Abstract

Because in practical studies, the right hand part of constant sum constraint from constant sum integer programming model is a random variable, we consider that the existence of the solutions for different values of right hand part of constant sum constraint can be considered a probabilistic problem. The aim of this paper is to study this problem for the simplest stochastic version for constant sum integer programming in which only named value is random variable.

Keywords: *Constant sum integer programming, combinatorial optimization, integer programming, stochastic optimization, stochastic integer programming, operational research*

ACM/AMS Classification: 90C05, 90C10, 90C15, 90C27

1. A short historical overview

A condition of constant sum type has appeared firstly when I wish to study a problem of mixing which it seems to need only a very simple model of *linear programming*. In this problem is specified that all variables are percentage, or else spoken, all variables have values in $[0, 1]$ and the sum of variables must be equal with 1.

This model has the form:

$$opt f = c \cdot x$$
$$\begin{cases} A \cdot x \leq b \\ \sum_{i=1}^{n} x_i = 1 \end{cases}$$
$$x_i \in [0, 1],\ i = 1, 2, \ldots\ldots, n$$

To solve this problem I used the instruments of linear programming.

For the above problem I consider that, in engineering practice, all variables given as $[0, 1]$ variables are not really in continuous interval $[0, 1]$ because engineers use, in general, no more then three decimals and so, from the interval $[0, 1]$, an accurate practical model must consider only the values of the form $p \cdot 10^-3$, where p is an integer between 0 and 1,000.

By scaling the model for the problem of mixing, the new model becomes an *integer programming model* (more precisely, a model of *combinatorial optimization*) in which it appears for the first time a condition in the form of constant sum condition. In this case, the right hand part of the constraint is a scaling factor, and also, this scaling value is the upper value for each variable.

Because this model is developed from a percentage problem, I call this type of model ***percentage programming*** ([1] and [2]) and in general form for this model I do not continue to speak about the scaling factor.

The general form I consider for *percentage programming model* (for scaling factor designated by S) is:

$$
\begin{array}{l}
opt\, f = c \cdot x \\
\begin{cases} A \cdot x \leq b \\ \sum\limits_{i=1}^{n} x_i = S \end{cases} \\
x_i \in \{0, 1, \ldots, S\}\,,\; i = 1, 2, \ldots\ldots, n
\end{array}
$$

To find the solution for *percentage programming model* I used the instruments of *integer programming.*

Trying to find a generalization for percentage model I realized that the condition to have the same value for right hand part of the constant sum constraint and for upper value for all variables can be eliminated.

In this way it is built a model belonging to *linear pure combinatorial optimization* which I call ***constant sum integer programming with uniform variables*** ([3]), firstly, because I consider that constant sum constraint is an important part for model solving, offering a good algorithm, and, secondly , I use the expression *with uniform variable* because all variables are finite and its have the values between 0 and a specified value Q.

This type of model has the following general form:

$$
\begin{array}{l}
opt\, f = c \cdot x \\
\begin{cases} A \cdot x \leq b \\ \sum\limits_{i=1}^{n} x_i = S \end{cases} \\
x_i \in \{0, 1, \ldots, Q\}\,,\; i = 1, 2, \ldots\ldots, n
\end{array}
$$

For solving *constant sum integer programming with uniform variable* I considered that the instruments of *integer programming* are not adequate any more, but in the same time I realized that the algorithms from *combinatorial optimization* are too slow, or very complicated.

This is why I tried to find a new algortihm, almost polynomial in complexity, based mainly on the advantage offered by the existance of constant sum constraint.

How the reader can see, all the models presented above belong to deterministic part of operational research and in this paper my desire is to go to the models which are stochastic.

This is the main point in which, studying another problem I see that a similar kind of model with *cconstant sum integer programming with uniform variable* can be applied to extend the results I have already found.

2. **Deterministic and stochastic constant sum integer programming models**

When I study the models for equipment replacement, after an approach by 0-1 programming that gives direct answer to my problem, I thought that the same problem can be approached indirectly, with the main goal the determination of the level of production on any existing equipments and the secondary goal the equipment replacement.

For the main goal of this approach I considered that it was proper to try the use of the last model presented in paragraph 1, but this model had a big limitation for practical application, namely, all equipment to have the same maximum level of production.

Elimination of this limitation produced a new model that generalized the model specified, and I called this one *constant sum integer programming with non-uniform variables* model or simpler, ***constant sum integer programming*** model ([4]).

So, I considered that any variable x_i had its own upper bound Q_i and the form for this new model is:

$$
\begin{aligned}
&opt f = c \cdot x \\
&\begin{cases} A \cdot x \le b \\ \sum\limits_{i=1}^{n} x_i = S \end{cases} \\
&x_i \in \{0, 1, \ldots, Q_i\}, \; i = 1, 2, \ldots\ldots, n
\end{aligned}
$$

For all the models specified above there exist characterizations and good algorithms (in complexity) to solve them.

The main problem for all this models is that in practical applications, they are only theoretical abstract models, because there are a lot of models parameters that are not exactly determined; in most cases they are determined with statistical instruments.

From a theoretical point of view, we can consider that the following models parameters are random:

- P – the right hand part in constant sum constraint is a random variable;
- Q_1, Q_2, $\cdots$,Q_n – the upper bounds for models variables are all random variables;
- $C = (c_1, c_2, \cdots, c_n)^t$ – the coefficient of objective function is a random vector;
- $B = (b_1, b_2, , b_m)^t$ – the right hand part of regular constraints system is a random vector;
- $A = (a_{ij})_{i=\overline{1,m}; j=\overline{1,n}}$ – the matrix of regular constraints system is a random matrix.

Any of the above possibilities or combinations of mentioned possibilities is more adequate in practical application, but considering it in a model makes the model to become a stochastic model and so we can speak, in our case, about stochastic constant sum integer programming.

3. **Determination of existence of solution for constant sum integer programming models**

In the rest of this paper we will consider the *constant sum integer programming* model in the form:

$$minf = c \cdot x$$
$$\begin{cases} A \cdot x \leq b \\ \sum_{i=1}^{n} x_i = S \end{cases}$$
$$x_i \in \{0, 1, \ldots, Q_i\}\,,\ i = 1, 2, \ldots\ldots, n$$

in which the coefficients of objective function are considered in increasing order (so, $c_1 \leq c_2 \leq \cdots \leq c_n$. This assumption does not restrict the generality because we can apply a permutation of indexes to all model so that this condition to be fulfilled).

First we must observe that, as it is specified in [5], with small variation for the value of P, let say ΔP, if solution of the model exists in P, it exists in $P - \Delta p$ and in $P + \Delta P$ and no more then three values are modified for the solutions in $P - \Delta P$ and $P + \Delta P$ in comparison with the solution in P.

Here, ΔP continues to be in non negative integer.

Because the space for the values of P is not a continuous space, we can not speak of convexity, but because it continues to be verified the convexity condition, we can speak about pseudo-convexity.

The results from [5] can be extended in this manner from the value of P where the solution of model exists, through right until we find that we reach the lower value possible for P or existing ΔP is zero, and through left until we find that we reach the higher value possible for P or existing ΔP is zero.

The above process makes possible to determine the values P_{low} and P_{high} so that for any P, $P_{low} \leq P \leq P_{high}$, the constant sum integer programming model with P the right hand part of constant sum constraint has solution.

In addition, we can give the following result which is important for the rest of our analysis:

Proposition 3.1. *Let us consider the constant sum integer programming model:*

$$minf = c \cdot x$$
$$\begin{cases} A \cdot x \leq b \\ \sum_{i=1}^{n} x_i = S \end{cases}$$
$$x_i \in \{0, 1, \ldots, Q_i\}\,,\ i = 1, 2, \ldots\ldots, n$$

in which we assume that the coefficients of objective function a in increasing order. If model has solution for $P = P_1$ *and model has solution for* $P = P_2$*, and* $P_1 < P_2$*, then model has solution for any* $P = T$*, with* $P_1 \leq T \leq P_2$*.*

Now let us consider the *constant sum integer programming* model and let us assume that we have given P_{min} and P_{max}, $P_{min} < P_{max}$, which determine the possible values for P, but we do not know if, and where, the model has solution for a given value of P, $P_{min} \leq P \leq P_{max}$.

The first operation to be realized on the model is to determine if there exists a value P for which the model has solution (answer to if question, if such value exists) and if the model has solution for a given P, to determine too the values P_{low} and P_{high} so that, for any P, $P_{low} \leq P \leq P_{high}$, the model has solution in P (answer to where question).

To do this operation we consider the following process in which we can consider that P_{low} and P_{high} were not determined if they have negative values:

Step 1. Let $P_{low} \leftarrow -1$, $P_{high} \leftarrow -1$, $i \leftarrow P_{min}$, $j \leftarrow P_{max}$, $P_c \leftarrow -1$, and $St \leftarrow \emptyset$.

Step 2. Solve model for P_{min}. If model has solution, then let $P_{low} \leftarrow P_{min}$ and $P_c \leftarrow P_{min}$ otherwise continue.

Step 3. Solve model for P_{max}. If model has solution then let $P_{high} \leftarrow P_{max}$ and $P_c \leftarrow P_{max}$ otherwise continue.

Step 4. If $P_{low} \neq -1$ and $P_{high} \neq -1$, then STOP ***bounds are determined***, otherwise continue.

Step 5. If $P_{low} \neq -1$ then goto step 16 otherwise if $P_{high} \neq -1$ then goto step 11 otherwise continue.

Step 6. Let $k \leftarrow [(i+j)/2]$ ($[\cdot]$ reprezent here and elsewere in the process the integer part of the number designated by $\cdot$).

Step 7. Solve model for k. If model has solution then let $P_c \leftarrow k$ and goto step 11 otherwise continue.

Step 8. If $k = i + 1$ then goto step 10 otherwise continue.

Step 9. $St \leftarrow$***push***(k, j), let $j \leftarrow k$, and goto step 6.

Step 10. If $St = \emptyset$ then STOP ***model if fully not admissible***, otherwise ***pop***$(i, j) \leftarrow St$ and goto step 6.

Step 11. Let $j \leftarrow P_c$.

Step 12. If $j = i + 1$ then let $P_{low} \leftarrow j$ and goto step 4 otherwise continue.

Step 13. Let $k \leftarrow [(i + j)/2]$.

Step 14. Solve model for k. If model has solution then let $j \leftarrow k$ otherwise let $i \leftarrow k$.

Step 15. Goto step 11.

Step 16. Let $i \leftarrow P_c$.

Step 17. If $j = i + 1$ then let $P_{high} \leftarrow i$ and goto step 4 otherwise continue.

Step 18. Let $k \leftarrow [(i + j)/2]$.

Step 19. Solve model for k. if model has solution then let $i \leftarrow k$ otherwise let $j \leftarrow k$.

Step 20. Goto step 17.

The reader can observe that this process, which solves the problem, acts like a binary search and in worst case the number of constant sum integer programming models solves is equal with $P_{max} - P_{min} + 1$, and because the algorithm for solving one constant sum integer programming models has the complexity $nlog_2n$, where n is the number o variable in model, it results that the above process is an algorithm and its complexity is equal with

$$(P_{max} - P_{min} + 1)nlog_2n.$$

We can resume the above analysis with the following result:

Theorem 3.1. *Let consider two given integers P_{min} and P_{max}, with $P_{min} < P_{max}$, and the family of constant sum integer programming models:*

$$\begin{array}{l} min f = c \cdot x \\ \left\{ \begin{array}{l} A \cdot x \leq b \\ \sum\limits_{i=1}^{n} x_i = S \end{array} \right. \\ x_i \in \{0, 1, \ldots, Q_i\}, \; i = 1, 2, \ldots \ldots, n \end{array}$$

where $P_{min} \leq P \leq P_max$ and we assume that the coefficients of objective function a in increasing order.

The process described above determines the domain of P in which the family of constant sum integer programming models has solutions or it gives the answer ***no model in the family has solution****. The process is an algorithm and the algorithm has complexity $O((P_{max} - P_{min} + 1)nlog_2n)$, where the maximum number of steps is realized when the answer of algorithm is* ***no model in the family has solution****.*

4. **Probabilities for existence of solution for stochastic constant sum integer programming with random constant sum constraint**

Let us consider, as it is in practical application of equipment replacements, that P (firstly considered as necessary level of production which satisfied the market requests) is a random variable specified by:

$$P : \begin{pmatrix} 0 & 1 & \cdots & P_0 & P_0+1 & P_0+2 & \cdots & P_0+t & P_0+t+1 & \cdots \\ 0 & 0 & \cdots & 0 & p_1 & p_2 & \cdots & p_t & 0 & \cdots \end{pmatrix}$$

with $p_1 + p_2 + \cdots + p_t = 1$.

Let us assume that the values $c_1, c_2, \cdots, c_n$ are given in increasing order and we give the constant sum integer programming model:

$$\begin{array}{l} min f = c \cdot x \\ \left\{ \begin{array}{l} A \cdot x \leq b \\ \sum\limits_{i=1}^{n} x_i = S \end{array} \right. \\ x_i \in \{0, 1, \ldots, Q_i\}, \; i = 1, 2, \ldots \ldots, n \end{array}$$

Because in this model P is a random variable, this model becomes a stochastic model and it represents the simplest version of constant sum integer programming model in stochastic form.

We can assume that, due to economic reason, we give the integer value P_{min} and P_{max}, $P_{min} < P_{max}$, and we specified that values for P are such as $P_{min} \leq P \leq P_{max}$.

We can consider three situations which appear for this model for a given value of P, namely:

- the model has solution;
- the model has not solution;
- it is not economically reasonable to solve the model.

In this section we wish to determine the probabilities for the three situations indicated above.

First let us consider the set

$$I_{ner} \left\{ i | P_0 + i < P_{min},\ 1 \leq i \leq t \right\} \bigcup \left\{ i | P_0 + i > P_{max},\ 1 \leq i \leq t \right\}$$

which represents the set of indexes for the values of P with non zero probabilities and outside the domain considered when problem is given. It is clear that for any $i \in I_{ner}$ the problem is considered as not reasonable to solve because it does not fulfill other type of conditions imposed by manufacturer.

We suppose that $\sum_{i \in \emptyset} p_i = 0$. Now we can give

$$P_{ner} = \sum_{i \in I_{ner}} p_i$$

which represents the indexes of probability of problems for which is considered that is not economicalyl reasonable to solve the model. This probability is certainly zero if manufacturer considers in solving the model all the value of P given by a study of market requests about the product.

Now, to determine the other two probabilities, we must apply the above algorithm to determine the sub-domain of P in which the model has solution.

Applying the algorithm, this either gives the answer ***no models has solution*** (for P, $P_{min} \leq P \leq P_{max}$) and in this case we can consider the sets:

$$I_{ns} = \left\{ i | P_{min} \leq P_0 + i \leq P_{max},\ 1 \leq i \leq t \right\},$$

and

$$I_s = \emptyset,$$

or it gives the value P_{low} and P_{high} so that for any P, $P_{low} \leq P \leq P_{high}$, the model has solution and in this case we can consider the sets:

$$I_s = \left\{ i | P_{low} \leq P_0 + i \leq P_{high},\ 1 \leq i \leq t \right\},$$

and

$$I_{ns} = \left\{ i | P_{min} \leq P_0 + i < P_{low},\ 1 \leq i \leq t \right\} \bigcup$$
$$\bigcup \left\{ i | P_{high} < P_0 + i \leq P_{max}, 1 \leq i \leq t \right\}.$$

The above sets represent:

- I_{ns} – the set of indexes of probability for the values for which the model has no solution;
- I_s – the set of indexes of probability for the values for which the model has solution.

Now it is obvious that:

- The probability for *model has solution* is given by:

$$P_s = \sum_{i \in I_s} p_i$$

- The probability for *model has no solution* is given by:

$$P_{ns} = \sum_{i \in I_{ns}} p_i$$

It is easy to verify that

$$I_{ner} \bigcup I_{ns} \bigcup I_s = \{1, 2, \ldots, t\}$$

and so

$$\begin{aligned} P_{ner} + P_s + P_{ns} &= \textstyle\sum_{i \in I_{ner}} p_i + \sum_{i \in I_s} p_i + \sum_{i \in I_{ns}} p_i \\ &= \textstyle\sum_{I_{ner} \bigcup I_{ns} \bigcup I_s} p_i \\ &= \textstyle\sum_{i=1}^{t} p_i = 1, \end{aligned}$$

which means that the three possibilities specified above for the model characterize completely the stochastic constant sum integer programming model with random constant sum constraint.

References

1. Bârză, S., *Programare procentuală: formulări şi proprietăţi*, Revista de Informatică, nr. 1, 2004, Inforec, Bucureşti.
2. Bârză, S., *Programare procentuală: proprietăţi ale soluţiilor*, Revista de Informatică, nr. 1, 2004, Inforec, Bucureşti.
3. Bârză, S., *Modele întregi cu sumă constantă şi variabile cu domeniu uniform*, Analele Universităţii Spiru Haret, Seria Matematică-Informatică, Vol. II, pp. 35-40, Bucureşti, 2006.
4. Bârză, S., *Constant Sum Integer Programming with Non Uniform Defined Variables*, International Conference on Operational Research for Development ICORD-V, Jamshedpur, India, December 2005, published in Proceedings of International Conference on Operational Research for Developments ICORD-V, Vol. II, pp. 255-262, India 2005.

5. Bârză, S., *Variation of Constant Sum Constraint for Integer Programming with Non Uniform Defined Variables*, International Conference APMOD 2008, Bratislava, May 2008, published in Analele Universităţii Spiru Haret, Seria Matematică-Informatică, Vol.IV, pp. 23-28, Bucureşti, 2008.
6. Bârză, S., *Modele cantitative pentru înlocuirea utilajelor*, Teză de doctorat, Academia de Studii Economice, Facultatea de Cibernetică, Statistică şi Informatică Economică, Bucureşti, decembrie 2005.
7. Goemans M.X., *Semidefinite Programming and Combinatorial Optimization*, Doc. Math. Extra Volume ICM, 1998, 657-666.
8. Goemans M.X., Rendl F., *Semidefinite Programming in Combinatorial Optimization*, November 1999.
9. Hoffman K.L., *Combinatorial Optimization: Current Successes and Directions for the Future*, Journal of Computational and Applied Mathematics 124 pp. 341, 2000.
10. Nemhauser G.L., Wolsey L.A., *Integer and combinatorial optimization*, John Wiley & Sons Inc, New York, 1999.