J. Bangladesh Acad. Sci., Vol. 44, No. 1, 13-31, 2020
DOI: https://doi.org/10.3329/jbas.v44i1.48560

A HEURISTIC ALGORITHM FOR SOLVING INTEGER LINEAR PROGRAMMING PROBLEM AND UNVEILING THE APPLICATIONS

ABIR SUTRA DHAR ** AND H. K. DAS*

Department of Mathematics, University of Dhaka, Dhaka-1000, Bangladesh

ABSTRACT

There is a growing need for integer solutions in industries, production units, etc. Specifically, there is an increasing demand to develop precise methods for solving integer-programming problems (IPPs). In this paper, we propose a new algorithm for solving IPPs in a general form by combining two decomposition techniques: Benders decomposition (BD) and decomposition-based pricing methods (DBP). Moreover, we generate some conditions for solving problems having either infeasible or unbounded solutions. In addition, we present an application and evaluation of a solution method for solving IPPs, while also giving a brief description of the different classical decomposition methods, namely the Dantzig-Wolfe decomposition (DWD), decomposition-based pricing (DBP), Benders decomposition (BD), and recently proposed improved decomposition (ID) methods for solving IPPs.We also discuss the use of the decomposition methods for solving IPPS to develop a heuristic algorithm, describe the limitations of the classical algorithms, and present extensions enabling its application to a broader range of problems. To illustrate the decomposition procedures, we will provide corresponding models and numerical results for two standard mathematical programs: the Fixed Charge Problem (FCP) and the Facility Location Problem (FLP). Our findings from this study suggest that our algorithm produces the most efficient computational solutions of IPPs.

Keywords: Integer linear programming, heuristic algorithm, Dantzig-Wolfe decomposition, decomposition-based pricing, Benders decomposition, fixed charge problem, facility location problem.

INTRODUCTION

Recent years have witnessed widespread use of integer programming problems to address many real life optimization problems. These problems have been evolved due to the implementation of linear programming (LP) when scientists realized that there was a need for certain models, especially models related to business problems, to have integer solutions. The integer variables refer to particular items, which are indivisible (number of machines, vehicles and others), and the continuous variables reflect mainly the estimates of price, time and other divisible objects. Nowadays Integer Linear Problems (ILP) are used in applications oriented towards decision making in many industrial and business activities. This is due not only to the increased computing power of modern computers, but also to improved ILP solvers, which enable easier formulation and solution of these models. So, ILPs have been around for a long time. In 1960, George Dantzig was a major contributor to the decomposition technique which was presented for linear program that permits the problem to be solved by alternate solutions of linear sub-programs representing its several parts and a coordinating program that is obtained from the parts by linear transformations. In recent years, researchers have used IPPs to solve optimization

*Corresponding author:<*hkdas_math@du.ac.bd; **abirsd07@gmail.com>.

problems in industry and business. This is due not only to the increased computing power of modern computers, but also as a result of improved methods to solve IPPs. In a typical setting, a LP problem in which all of the variables must take integer values is called an IPP. A number of integer algorithms have been developed to solve the various types of IPPs. The cutting plane method was developed at the end of the 1950's by Gomory for solving integer programming and integer programming problems using the simplex method. Once the simple Cutting plane method does not work for solving IPPs, then in the mid-1990s Gérard Cornuéjols and co-workers showed them to be very effective in combination with branch-and-bound(called branch-and-cut) and ways to overcome numerical instabilities. This approach works by branching on variable restrictions for solving IPPs, which is largely used method in practice, is neither efficient for solving an IPPs. For more general problems, a combination of cutting planes with a branch and bound procedure was designed by Crowder, Johnson and Padberg in 1983 which was able to solve large scale IPPs with sparse constraint matrix.

Moreover, Benders' decomposition, branch and cut, branch and bound, column generation and Dantzig–Wolfe decomposition procedures, developed after World War II, are useful for solving those types of problems. In 2014, Islam, Hasan, and Das developed a new DBP procedure to filter unnecessary decision ingredients from large scale IPPs. While many authors may have developed new decomposition procedures for solving IPPs, they did not account for infeasible solutions or bounded solutions. A few of their procedures are based on DWD, DBP, or BD methods. Many of them did not use any application of IPPs. While these methods are successful, we will figure out which methods are best for solving an MIP. In addition, the optimality condition described by these methods requires more iteration. In this paper, we developed a successful and time-consuming method to solve IPPs. Despite experimenting with different alternatives for IPPs, we then developed an effective decomposition method.

In this paper, there are two principal motivations for solving IPPs. First, we develop a new algorithm for solving IPPs by using both Benders decomposition and DBP. The approach is similar to Benders decomposition, but our algorithm involves using the dual value from DBP. Second, we illustrate the algorithm by considering five examples, in particular two applications of IPPs and comparing our algorithm with the other methods.

The remainder of the paper is organized as follows. First, we briefly discuss some existing classical decomposition methods for solving IPPs. Second, we develop a new algorithm illustrated by a flowchart. Third, we compare our algorithm with the other standard decomposition methods for solving IPPs. Finally, we show that our method is effective for solving IPPs along with Benders decomposition method.

MATERIALS AND METHODS

This section demonstrates some existing methods for solving the IPPs along with our method. Furthermore, we illustrate our algorithm by a flowchart for better insight into the phenomenon.

Dantzig–Wolfe decomposition Method

The way of exploiting the block-angular structure is Dantzig-Wolfe decomposition, which was invented by Dantzig and Wolfe in 1961. Moreover, this method is very closely connected to column generation and they are often used interchangeably.

The problem being solved is split into two problems:

(i) The master problem
(ii) The sub problem

Step-1: The master problem is solved. From this solution, we are able to obtain dual prices for each of the constraints in the master problem and this information is then utilized in the objective function of the sub problem.

Step-2: The sub problem is solved. This variable is then added to the master problem and continued Step1. the process is repeated until no variables with negative reduced cost are identified.

Decomposition Based Pricing Method

Decomposition-based pricing (DBP) for the efficient solution of integrated fishery planning problem: Mamer and McBride developed DBP for multicommodity flow problems. DBP procedure is summarized as follows.

Step-1: Relax complicating constraints by subtracting from objective function of the original problem. Decompose the whole problem into sub-problems and a master problem. Solve sub-problems and generate master problem by deleting those variables, which do not provide non-negative values from the original problem.

Step-2: Stop when sub problem value and master problem value become equal. Otherwise, repeat the previous steps.

Benders Decomposition Method

Benders decomposition is closely related to other decomposition methods for linear programming and having some relationships among Benders, Dantzig-Wolfe, and Lagrangian optimization. It is a solution method for solving certain large-scale optimization problems. Instead of considering all decision variables and constraints of a large-scale problem simultaneously, it partitions the problem into multiple smaller problems.

The problem being solved is split into two problems:

(i) The master problem
(ii) The sub problem

Step-1: We have chosen the complicating variable and the initial master problem is solved. If the sub problem is infeasible, artificial variables are included in the sub problem.

Step-2: After solving the sub problem, we got dual value. The optimal solution of sub problem and dual value are used in the master problem. Solving this master problem again. The process is repeated until no variables with negative reduced cost are identified.

Improved Decomposition Method

Due to the delayed column generation for solving large scale IPPs by DWD principle, in 2013 Istiaq and Hasan presented an Improved Decomposition algorithm depending on DWD principle for solving IPPs into the following way. The problem being solved is split into two problems:

(i) The master problem
(ii) The sub problem

Step-1: The master problem is solved. From this solution, we are able to obtain dual prices for each of the constraints in the master problem and this information is then utilized in the objective function of the sub problem.

Step-2: The sub problem is solved. This variable is then added to the master problem and continued Step1. The process is repeated until no variables with negative reduced cost are identified.

That is, the method is composed of three sub problems (which can be generalized for n sub problems) of an original problem and the master problem with the help of Lagrangian relaxation. Optimality holds when the value of the sum of the sub-problem will be equal to the master problem, $V(S_1) + V(S_2) + V(S_3) = V(M)$.

Therefore, picking up an initial value of the dual variables randomly the sub problem(s) is solved from which current solution of the sub problem is imported to create the master problem. Then master problem is solved and the optimality condition is tested. If the optimality condition does not hold, then the current dual value from the master problem is taken and imported this to update the sub problem(s) and continue the same process unless it meets the optimality condition. This method is almost same computational skill to find the optimal solution in IPPs. There are many BD extensions and researchers explore algorithmic enhancements for the BD extensions. Therefore, we now introduce a new algorithm in details as well as discuss by flowchart in the following Section by extending the idea of BD method.

Our Decomposition Approach

The heuristic algorithm is used to find non negative integer solutions in a general form. It is an algorithm that allows us to solve certain optimization problems very quickly. The algorithm can be used on any kind of optimization problem but requires a certain substructure within the problem to be efficient that can be demonstrated in the following way.

Basic idea of the algorithm

Here P1 means original problem.

Main problem

$$\begin{aligned} &(\text{P1})\text{minimize } p^T x + q^T y \\ &s.t. \quad Ay \geq b \\ &\quad Dx + Gy \geq h \\ &\quad x \geq 0, y \in S \end{aligned} \tag{1}$$

Since x is continuous and y is fixed, S is set continuous. Then original problem can be decoupled into a master problem (MP) and a sub problem (SP). p^T, q^T are cost coefficient, A, D, G are coefficient of constrain, b, h are our right-hand sides.

Master problem (MP1)

$$\begin{aligned} &\text{minimize } z_l \\ &s.t. \quad z_l - q^T y \geq 0 \\ &\quad Ay \geq b \\ &\quad y \in S \end{aligned} \tag{2}$$

z_l is our cost coefficient of master problem

Sub problem (SP1)

$$\begin{aligned} &\text{minimize } p^T x + q^T \hat{y} \\ &s.t. \quad A\hat{y} \geq b \\ &\quad Dx \geq h - G\hat{y} \\ &\quad x \geq 0, \hat{y} \in S \end{aligned} \tag{3}$$

$\hat{y}$ is solution of MP1

Sub problem (SP1-Dual)

Dual form of SP1,

$$\begin{aligned} &\text{maximize } (h - G\hat{y})^T v + b^T u \\ &s.t. \quad D^T v \leq p \\ &\quad Au \leq q \\ &\quad v, u \geq 0 \end{aligned} \tag{4}$$

v, u be dual variables

The Proposed Algorithm

Step 1. Solve MP1 and obtain solution given as $\hat{z}_l$ at $\hat{y}$. If MP1 is infeasible, the original problem P1 will have either no feasible solution or an unbounded solution. Stop the process. If not, then we go to Step 2.

Step 2. Generate sub problem by deleting those variables which do not provide non negative values from the master problem. We solve SP1 or SP1-Dual and obtain solution is $\hat{z}_u$ and get dual value. Then new optimal solution of SP1 is $z_{new} = \hat{z}_u + \hat{z}_l$. Where $\hat{z}_u$ is dual value and $\hat{z}_l$ is solution of MP1.

i. If $\left|\hat{z}_{new} - \hat{z}_l\right| \leq \varepsilon$ then stop the process. Otherwise, generate a new constrain $\hat{z}_l - q^T y - (\text{h} - \text{Gy})^T v^r \geq 0$ (feasibility cut) for MP2 where, v^r comes from the optimal solution of SP2. v^r is generated of optimal solution here r denote row.

ii. If SP1-Dual is unbounded, which means that SP1 is infeasible, then added a new cut $-(\text{h} - \text{Gy})^T w^r \leq 0$ (infeasible cut) for MP2. In this case, we will first calculate w^r form the infeasibility cut and then go to step 3. We use a new SP1 (5), feasibility check sub problem to calculate w^r in SP2. W^r is generated of optimal solution here r denotes row.

iii.

$$\begin{aligned} &\text{minimize } 1^T s \\ &s.t. Dx + Is \geq h - G\hat{y} \qquad (5) \\ &\qquad x \geq 0, s \geq 0 \end{aligned}$$

Here 1 is the unit vector and I is new vector ands is new variable

iv. If SP1-Dual is infeasible, the main problems P1 will not any feasible solution or an unbounded solution. Stop the process.

Step 3. Solve MP2 to obtain a solution $\hat{z}_l$ w.r.t. $\hat{y}$. The feasibility cut (second constraint) or the infeasibility cut (third constraint) is adding with MP2 which being discussed in step2.

$$\begin{aligned} &\text{Minimize } z_l \\ &s.t. \quad Ay \geq b \\ &\qquad z_l - q^T y - (\text{h} - \text{Gy})^T v^r \geq 0, i = 1,\ldots,n \qquad MP2 \\ &\qquad -(\text{h} - \text{Gy})^T w^r \geq 0, i = 1,\ldots n \\ &\qquad y \in S \end{aligned}$$

i. If MP2 is infeasible then main problem P1 must infeasible.

ii. If MP2 is unbounded so $z_l = \infty$ go to Step-2

Step-4. Then go to back Step-2 and solve Sub problem again. This process to be continuing until the master problem=sub problem.

Flowchart of the proposed Algorithm

In this section, we discuss our decomposition method by flowchart.

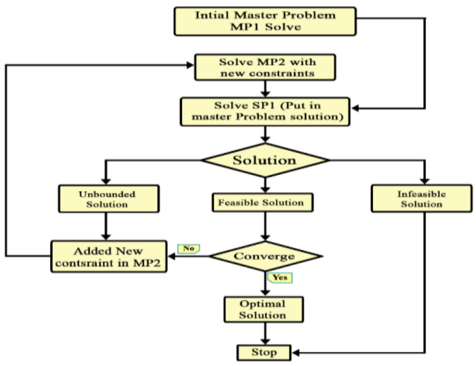

Fig. 1. Flowchart of the Algorithm

Research into the decomposition methods are not yet complete, as there are still many challenges and open questions. In addition, recent studies have presented various modified decomposition strategies though.

We conclude here the discussions of the decomposition algorithms. Next Section demonstrates how the prescribed decomposition procedures work in problem encountered in the FCP and the FLP and giving a numerical comparison.

RESULTS AND DISCUSSION

We performed experiments in this section to compare the execution of Benders' method, DWD, DBP, ID and the proposed decomposition methods on the two classical optimization problems: the FCP and the FLP and some other problems in this field and find the best algorithm for solving of IPPs.

The Fixed Charge Problem

Example 1

Gandhi Cloth Company is capable of manufacturing three types of clothing: shirts, shorts and pants. The manufacture of each type of clothing requires that Gandhi has the appropriate type of machinery available. The machinery needed to manufacture each type of clothing must be rented at the following rates: shirts machinery, 200$ per weeks; shorts machinery, 150$ per week; pants machinery, 100$ per week. The manufacture of each type of clothing are requires the amounts of cloth and labor shown in Table-B. Each week, 150 hours of labor and 160 sq yd of cloth are available. The variable unit cost and selling price for each type of clothing are shown in Table-A. This example is taken from Winston

Table 1. Data of Example-1

Table-A			Table-B		
Clothing Type	**Sales Price ($)**	**Variable Costs ($)**	**Clothing type**	**Labor (hours)**	**Cloth (square yards)**
Shirts	12	6	Shirts	3	4
Shorts	8	4	Shorts	2	3
Pants	15	8	Pants	6	4

Solution

Solution of Example 1 by Using DWD Method

Table 2. Result of Example-1by using DWD

It.#	Sub problem Solution		Master problem Solution		Dual Value
	x_i	$S_i(\text{v})$	θ_i	$M_i(\text{v})$	λ_i
1	$x_1 = x_2 = x_3 = 0$ $y_1 = y_2 = y_3 = 0$	4700	$\theta_1 = 1$	0	$\lambda_1 = 100$ $\lambda_2 = 200$
2	$x_1 = 40, x_2 = 53, x_3 = 25$ $y_1 = y_2 = y_3 = 1$	177	$\theta_1 = 0.618138, \theta_2 = 0.381862$	67.589	$\lambda_1 = 0$ $\lambda_2 = 0.422434$
3	$x_1 = x_2 = 0, x_3 = 25$ $y_1 = y_2 = 0, y_3 = 1$	96.12	$\theta_1 = 0, \theta_2 = 0, \theta_3 = 1$	75	$\lambda_1 = 0.451327$ $\lambda_2 = 0$
4	$x_1 = x_2 = 0, x_3 = 25$ $y_1 = y_2 = 0, y_3 = 1$	75	$\theta_1 = \theta_2 = \theta_3 = 0, \theta_4 = 1$	75	$\lambda_1 = 0.451327$ $\lambda_2 = 0$

Solution of Example 1 by Using DBP Method

Table 3. Result of Example-1by using DBP

It.#	Sub problem Solution		Master problem Solution		Dual value
	x_i *and* y_i	$S_i(\text{v})$	x_i *and* y_i	$M_i(\text{v})$	λ_i
1	$x_1 = x_2 = 0, x_3 = 25, y_1 = y_2 = 0, y_3 = 1$	60	$x_3 = 25, y_3 = 1$	75	$\lambda_1 = 0.1$
2	$x_1 = x_2 = 0, x_3 = 25, y_1 = y_2 = 0, y_3 = 1$	75	$x_3 = 25, y_3 = 1$	75	$\lambda_2 = 0$

Solution of Example 1 by Using BD Method

Table 4. Result of Example-1by using BD

It. #	Sub problem Solution		Master problem Solution		Dual value
	x_i *and* y_i	$S_i(\text{v})$	x_i	$M_i(\text{v})$	λ_i
1	$x_1 = 30, x_2 = 0, x_3 = 10$ $y_1 = y_2 = 0, y_3 = 1$	75	$x_1 = 30, x_2 = 0, x_3 = 10$	250	$\lambda_1 = 5, \lambda_2 = 4, \lambda_3 = 4$
2	$x_1 = x_2 = 0, x_3 = 25$ $y_1 = y_2 = 0, y_3 = 1$	75	$x_1 = 0, x_2 = 0, x_3 = 25$	75	$\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 4$

Solution of Example 1 by Using ID Method

Table 5. Result of Example-1 by using ID

It.#	Sub problem Solution		Master problem Solution		Dual value
	x_i	S_i(v)	θ_i	M_i(v)	λ_i
1	$x_1 = x_2 = x_3 = 0$ $, y_1 = y_2 = y_3 = 0$	4700	$\theta_1 = 1$	0	$\lambda_1 = 100, \lambda_2 = 200$
2	$x_1 = 40, x_2 = 53, x_3 = 25$ $y_1 = y_2 = y_3 = 1$	177	$\theta_1 = 0.618138,$ $\theta_2 = 0.381862$	67.589	$\lambda_1 = 0, \lambda_2 = 0.42$
3	$x_1 = x_2 = 0, x_3 = 25$ $y_1 = y_2 = 0, y_3 = 1$	96.12171	$\theta_1 = 0, \theta_2 = 0, \theta_3 = 1$	75	$\lambda_1 = 0.45, \lambda_2 = 0$
4	$x_1 = x_2 = 0, x_3 = 25$ $y_1 = y_2 = 0, y_3 = 1$	75	$\theta_1 = \theta_2 = \theta_3 = 0, \theta_4 = 1$	75	$\lambda_1 = 0.45, \lambda_2 = 0$

Solution of Example 1 by Using our Algorithm

Table 6. Result of Example-1by using Our Procedure

It. #	**Sub problem Solution**		**Master problem Solution**		**Dual value**
	x_i *and* y_i	S_i(v)	$\boldsymbol{x_i}$	M_i(v)	v^i
1	$x_1 = x_2 = 0, x_3 = 25$ $, y_1 = y_2 = 0, y_3 = 1$	75	$x_1 = 30, x_2 = 0, x_3 = 10$	250	$v^1 = 5, v^2 = 4, v^3 = 4$
2	$x_1 = x_2 = 0, x_3 = 25$ $, y_1 = y_2 = 0, y_3 = 1$	75	$x_1 = 0, x_2 = 0, x_3 = 25$	75	$v^1 = 0, v^2 = 0, v^3 = 4$

Example 2

Mr. Amit has been approached by three telephone companies to subscribe to their long distance service in the United States. MaBelll will charge a flat $16 per month plus $.25 a minute. PaBell will charge $25 a month but will reduce the per minute cost to $.22. As for BabyBell, the flat monthly charge is $18, and the cost per minute is $21. I usually make an average of 200 minutes of long-distance calls a month. Assuming that I do not pay the flat monthly fee unless I make calls and that I can apportion my calls among all three companies as I please, how should I use the three companies to minimize my monthly telephone bill? This example is taken from Taha.

Solution

Solution of Example 2 by Using DWD Method

Table 7. Result of Example-2by using DWD

It. #	Sub problem Solution		Master problem Solution		Dual Value
	x_i *and* y_i	S_i(v)	θ_i	M_i(v)	λ_i
1	$x_1=x_2=0, x_3=200, y_1=y_2=y_3=0$	50	$\theta_1=1$	0	$\lambda_1=0.02$
2	$x_1=0, x_2=0, x_3=200, y_1=y_2=y_3=0$	50	$\theta_1=1, \theta_2=1$	60	$\lambda_1=0$
3	$x_1=x_2=0, x_3=200, y_1=y_2=0, y_3=1$	60	$\theta_1=1, \theta_2=0, \theta_3=1$	60	$\lambda_1=0.25$

Solution of Example 2 by Using DBP Method

Table 8. Result of Example-2by using DBP

It. #	Sub problem Solution		Master problem Solution		Dual value
	x_i and y_i	S_i(v)	x_i *and* y_i	M_i(v)	λ_i
1	$x_1=0, x_2=200, x_3=0,$ $y_1=y_2=y_3=0$	22	$x_2=200$	50	$\lambda_1=0.1$
2	$x_1=x_2=0, x_3=200, y_1=y_2=0,$ $y_3=1$	60	$x_3=200, y_3=1$	60	$\lambda_1=0$

Solution of Example 2 by Using BD Method

Table 9. Result of Example-2by using BD

It. #	Sub problem Solution		Master problem Solution		Dual value
	y_i	S_i(v)	x_i	M_i(v)	λ_i
1	$y_1=y_2=0, y_3=1$	60	$x_1=x_2=0, x_3=200$	42	$\lambda_1=0$
2	$y_1=y_2=0, y_3=1$	60	$x_1=x_2=0, x_3=200$	60	$\lambda_1=0, \lambda_2=0, \lambda_3=0.09$

Solution of Example 2 by Using ID Method

Table 10. Result of Example-2 by using ID

It. #	Sub problem Solution		Master problem Solution		Dual Value
	x_i *and* y_i	S_i(v)	θ_i	M_i(v)	λ_i
1	$x_1 = x_2 = 0, x_3 = 200, y_1 = y_2 = y_3 = 0$	50	$\theta_1 = 1$	0	$\lambda_1 = 0.02$
2	$x_1 = 0, x_2 = 0, x_3 = 200, y_1 = y_2 = y_3 = 0$	50	$\theta_1 = 1, \theta_2 = 1$	60	$\lambda_1 = 0$
3	$x_1 = x_2 = 0, x_3 = 200, y_1 = y_2 = 0, y_3 = 1$	60	$\theta_1 = 1, \theta_2 = 0, \theta_3 = 1$	60	$\lambda_1 = 0.25$

Solution of Example 2 by Using Our Procedure

Table 11. Result of Example-2 by using Our Procedure

It. #	Sub problem Solution		Master problem Solution		Dual value
	x_i *and* y_i	S_i(v)	x_i	M_i(v)	v_i
1	$x_1 = x_2 = 0, x_3 = 200$ $y_1 = y_2 = 0, y_3 = 1$	60	$x_1 = x_2 = 0, x_3 = 200$	42	$v_1 = v_2 = v_3 = 0$
2	$x_1 = x_2 = 0, x_3 = 200$ $y_1 = y_2 = 0, y_3 = 1$	60	$x_1 = x_2 = 0, x_3 = 200$	60	$v_1 = v_2 = 0, v_3 = 0.09$

The Facility Location Problem

We again explain here how DWD, DBP, BD, improved decomposition and the proposed decomposition methods work, and how those can be used on a facility location problem in order to illustrate the mechanics of the algorithms.

Example 3

This example will use 3 possible factories and 5 possible customers. The following table lists the c_{ij} 's as well as the f_i 's.

Table 12. Facilities Location Problem

Factory	Customers					Fixed costs($)
	1	2	3	4	5	
1	2	3	4	5	7	2
2	4	3	1	2	6	3
3	5	4	2	1	3	3

We find out how many factories are open at minimum cost? This example is taken from Hooker.
Solution

Table 13. Result of Example-3using DWD

It. #	Sub problem Solution		Master problem Solution		Dual value
	x_{ij} *and* y_i	$S_i(\text{v})$	θ_i	$M_i(\text{v})$	λ_i
1	$x_{31}=x_{32}=x_{13}=x_{14}=x_{15}=x_{21}=x_{22}=x_{23}=x_{24}=x_{25}=0$, $x_{11}=x_{12}=x_{33}=x_{34}=x_{35}=1, y_1=1, y_2=0, y_3=1$	16	$\theta_1=1$	16	$\lambda_1=5$

Using DBP Method

Table 14. Result of Example-3using DBP

It. #	Sub problem Solution		Master problem Solution		Dual value
	x_{ij} *and* y_i	$S_i(\text{v})$	x_i *and* y_i	$M_i(\text{v})$	λ_i
1	$x_{31}=x_{32}=x_{13}=x_{14}=x_{15}$ $=x_{21}=x_{22}=x_{23}=x_{24}=x_{25}=0$, $x_{11}=x_{12}=x_{33}=x_{34}=x_{35}=1$ $y_1=1, y_2=0, y_3=1$	15	$x_{11}=x_{12}=x_{33}=x_{34}=1$ $x_{35}=1, y_1=y_3=1$	16	$\lambda_1=2$
2	$x_{31}=x_{32}=x_{13}=x_{14}=x_{15}$ $=x_{21}=x_{22}=x_{23}=x_{24}=x_{25}=0$, $x_{11}=x_{12}=x_{33}=x_{34}=x_{35}=1$ $y_1=1, y_2=0, y_3=1$	16	$x_{11}=x_{12}=x_{33}=x_{34}=1$ $x_{35}=1, y_1=y_3=1$	16	$\lambda_1=0$

Using BD Method

Note that, $i \in O(\overline{y})$ =open factories and $j \in C(\overline{y})$ = closed factories and v_j are the dual variables associated with the demand constraints and w_{ij} are the dual variables associated with the setup constraints. We choose open factories 3, closed factories 1 and 2.

Table 15. Result of Example-3 by using BD

It.#	Sub problem Solution		Master problem Solution	
	v_j and w_{ij}	$S_i(\text{v})$	y_i	$M_i(\text{v})$
1	$v_j=(2,3,4,5,7), w_{1j}=(2,0,0,0,0), w_{2j}=w_{3j}=0$	23	$y_1=0, y_2=y_3=1$	10
2	$v_j=(4,3,1,1,3), w_{1j}=(2,0,0,0,0), w_{2j}=w_{3j}=0$	18	$y_1=0, y_2=0, y_3=1$	15
3	$v_j=(5,4,2,1,3), w_{1j}=(3,1,0,0,0)$, $w_{2j}=(1,1,1,0,0), w_{3j}=0$	16	$y_1=1, y_2=0, y_3=1$	16

Using ID Method

Table 16. Result of Example-3 by using BD

It. #	Sub problem Solution	Master problem Solution			Dual value
	x_{ij} *and* y_i	S_i(v)	θ_i	M_i(v)	λ_i
1	$x_{31}=x_{32}=x_{13}=x_{14}=x_{15}=x_{21}=x_{22}=x_{23}=x_{24}=x_{25}=0$, $x_{11}=x_{12}=x_{33}=x_{34}=x_{35}=1, y_1=1, y_2=0, y_3=1$	16	$\theta_1=1$	16	$\lambda_1=5$

Using Our Procedure
We choose open factories 3, closed factories 1 and 2.

Table 17. Result of Example-3by using Our Procedure

It.#	Sub problem Solution		Master problem Solution		Dual value
	x_{ij} *and* y_i	S_i(v)	y_i	M_i(v)	v_j *and* w_{ij}
1	$x_{31}=x_{32}=x_{13}=x_{14}=x_{15}=$ $x_{21}=x_{22}=x_{23}=x_{24}=x_{25}=0$ $,x_{11}=x_{12}=x_{33}=x_{34}=x_{35}=1$	16	$y_1=1, y_2=0, y_3=1$	16	$v_j=(5,4,2,1,3)$, $w_{1j}=(3,1,0,0,0)$, $w_{2j}=(1,1,1,0,0)$ $w_{3j}=0$

Example 4

Steel manufactures two types of steel (steel 1and steel 2) at two locations (plant 1 and plant 2). Three resources are needed to manufacture a ton of steel: iron, coal and blast furnace time. The two plants have different types of furnaces, so the resources needed to manufacture a ton of steel depend on the location (see table 1). Each plant has its own coal mine. Each day, 12 tons of coal are available at plant 1 and 15 tons at plant 2. Coal cannot be shipped between plants. Each day, plant 1 has 10 hours of blast furnace time available, and at plant 2 has 4 hours available. Iron ore is mined in a mine located midway between the two plants: 80 tons of iron are available each day. Each ton of steel 1 can be sold for \$ 170/ton, and each ton of steel 2 can be sold for \$ 160/ton. All steel that is sold is shipped to a single customer. It costs \$ 80 to ship a ton of steel from plant 1, and \$100 a ton from plant 2. Assuming that the only variable costs is the shipping cost, formulate and solve the IPP to maximize Steelco’s revenue less shipping cost. This example istaken from Winston.

Table 18. Steelco Manufacturing Problem

Product (tons)	Iron required (tons)	Coal required (tons)	Blast furnace time required(hours)
Steel 1 at plant 1	8	3	2
Steel 2 at plant 1	6	1	1
Steel 1 at plant 2	7	3	1
Steel 2 at plant 2	5	2	1

Solution

Solution of Example-4 by using DWD

Table 19. Result of Example-4 by using DWD

It.#	Sub problem Solution		Master problem Solution		Dual value
	x_i	S_i(v)	θ_i	M_i(v)	λ_i
1	$x_1 = x_2 = 0$ $x_3 = x_4 = 0$	8000	$\theta_1 = 1$	0	$\lambda_1 = 100$
2	$x_1 = 0, x_2 = 10$ $, x_3 = 4, x_4 = 0$	1080	$\theta_1 = 0.090909$ $, \theta_2 = 0.909091$	981.8182	$\lambda_2 = 0$
3	$x_1 = 0, x_2 = 10$ $, x_3 = 0, x_4 = 0$	1045.455	$\theta_1 = 0, \theta_2 = 0.714286$ $, \theta_1 = 0.285714$	1000	$\lambda_3 = 12.272727$
4	$x_1 = 0, x_2 = 10$ $, x_3 = 0, x_4 = 0$	1040	$\theta_1 = 0, \theta_2 = 0, \theta_3 = 0, \theta_4 = 1$	1040	$\lambda_4 = 10$

Solution of Example 4 by using DBP

At first we choose a dual value $\lambda_1 = 0$ then moderated dual value put in master problem, this procedure continue until master problem and sub problem are equal.

Table 20. Result of Example-4 by using DBP

It.#	Sub problem Solution		Master problem Solution		Dual value
	x_i	S_i(v)	θ_i	M_i(v)	λ_i
1	$x_1=0, x_2=10, x_3=4, x_4=0$	1080	$x_1=0, x_2=10, x_3=2, x_4=0$	1000	$\lambda_1=0$
2	$x_1=0, x_2=10, x_3=0, x_4=4$	1040	$x_1=0, x_2=10, x_3=0, x_4=4$	1040	$\lambda_2=10$

Table 21. Result of Example-4 by using BD

It. #	Sub problem Solution		Master problem Solution		Dual value
	u_i	S_i(v)	x_i	M_i(v)	λ_i
1	$u_1=12, u_2=0, u_3=0$	240	$x_1=0, x_2=10, x_3=0, x_4=0$	800	$\lambda_1=0$
2	$u_1=12, u_2=0, u_3=0$	1040	$x_1=0, x_2=10, x_3=0, x_4=0$	1040	$\lambda_2=12$

Table 22. Result of Example-4 by using ID

It. #	Sub problem solution		Master problem solution		Dual value
	x_i	S_i(v)	θ_i	M_i(v)	λ
1	$x_1=x_2=x_3=x_4=0$	8000	$\theta_1=1$	0	100
2	$x_1=x_4=0, x_2=10,$ $x_3=4$	1080	$\theta_1=.090909,$ $\theta_2=.90909$	981.81	0
3	$x_1=x_3=x_4=0,$ $x_2=10$	1045.45	$\theta_1=0, \theta_2=0.714286$ $,\theta_3=0.285714$	1000	12.272727
4	$x_1=x_3=0, x_2=10,$ $x_4=4$	1040	$\theta_1=\theta_2=\theta_3=0, \theta_4=1$	1040	0

Table 23. Result of Example-4 by using Our Algorithm

It. #	**Sub problem Solution**		**Master problem Solution**		**Dual value**
	x_i	S_i(v)	$\hat{x}_i$	M_i(v)	v^r
1	$x_2 = 10, x_3 = 2, x_4 = 0$	1040	$\hat{x}_1 = 0, \hat{x}_2 = 10$	800	$v^1 = 0$
2	$x_2 = 10, x_3 = 0, x_4 = 4$	1040	$\hat{x}_1 = 0, \hat{x}_2 = 10$	1040	$v^2 = 12$

Example 5

$$\begin{aligned} \text{maximmize} \quad & z = 7x_1 + 5x_2 + 3x_3 \\ \text{subject to} \quad & x_1 + 2x_2 + x_3 \le 10 \\ & x_1 + x_2 \le 5 \\ & x_1 \le 3 \\ & 2x_2 + x_3 \le 8 \\ & x_1, x_2, x_3 \ge 0 \quad \text{integer} \end{aligned}$$

Solution

Solution of Example 5 by using DWD

We now solve this problem by using DWD in following table.

Table 24: Result of Example-5 by using DWD

It. #	**Sub problem Solution**		**Master problem Solution**		**Dual value**
	x_i	S_i(v)	θ_i	M_i(v)	
1	$x_1 = 3, x_2 = x_3 = 0$	56	$\theta_1 = 1$	21	5
2	$x_1 = 3, x_2 = 3, x_3 = 2$	42	$\theta_1 = 0.125, \theta_2 = 0.875$	39.375	0
3	$x_1 = 3, x_2 = 0, x_3 = 5$	41.25	θ_1=0,θ_2=0.666667,θ_3=0.333333	40	2.625
4	$x_1 = 3, x_2 = 2, x_3 = 3$	40	θ_1=0,θ_2=0.666667,θ_3=0.333333	40	2

Solution of Example 5 by using DBP Method

We now solve this problem by using DWD in following table.

Table 25. Result of Example-5 by using DBP

It.#	Sub problem Solution		Master problem Solution		Dual value λ_i
	x_i	$S_i(\mathrm{v})$	x_i	$M_i(\mathrm{v})$	
1	$x_1=0, x_2=0, x_3=0$	1000	$x_1=0, x_2=0, x_3=0$	0	$\lambda_1=100$
2	$x_1=3, x_2=2, x_3=3$	40	$x_1=3, x_2=2, x_3=3$	40	$\lambda_2=0$

Solution of Example 5 by using BD Method

We now solve this problem by using DWD in following table.

Table 26. Result of Example-5 by using BD

It. #	**Sub problem Solution**		**Master problem Solution**		**Dual value** λ_i
	x_i	$S_i(\mathrm{v})$	x_i	$M_i(\mathrm{v})$	
1	$x_2=2, x_3=3$	40	$x_1=3$	21	$\lambda_1=2, \lambda_2=1, \lambda_3=0$
2	$x_2=2, x_3=3$	40	$x_1=3$	40	$\lambda_1=0, \lambda_2=0, \lambda_3=0$

Solution of Example 5 by using ID Method

Table 27. Result of Example-5 by using ID

It. #	**Sub problem Solution**		**Master problem Solution**		**Dual value** λ
	x_i	$S_i(\mathrm{v})$	θ_i	$M_i(\mathrm{v})$	
1	$x_1=3, x_2=x_3=0$	56	$\theta_1=1$	21	5
2	$x_1=3, x_2=3, x_3=2$	42	$\theta_1=0.125, \theta_2=0.875$	39.375	0
3	$x_1=3, x_2=0, x_3=5$	41.25	$\theta_1=0, \theta_2=0.666667, \theta_3=0.333333$	40	2.62
4	$x_1=3, x_2=2, x_3=3$	40	$\theta_1=0, \theta_2=0.666667, \theta_3=0.333333$	40	2

Solution of Example 5 by using Our Procedure

Table 28. Result of Example-5 by using Our Procedure

It. #	Sub problem Solution		Master problem Solution		Dual value
	x_i	$S_i(\text{v})$	x_i	$M_i(\text{v})$	v^r
1	$x_2 = 2, x_3 = 3$	40	$x_1 = 3$	21	$v^1 = 0$
2	$x_2 = 2, x_3 = 3$	40	$x_1 = 3$	40	$v^1 = 2$

Comparison and Findings

Our aim in this section is to find the best algorithm for solving of IPPs. We present the graphical comparison of our decomposition procedure with DWD, DBP, ID and BD. In the following figures, blue color indicates the iteration number and horizontal axis illustratesthe name of decomposition procedures. By these experiments, we conclude that our algorithm produce the most efficient computational solutions of IPPs.

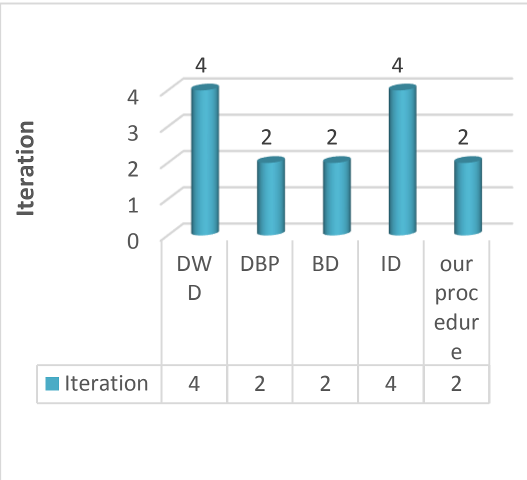

Fig. 2. Iteration Comparison of Decomposition Procedures for Example-1

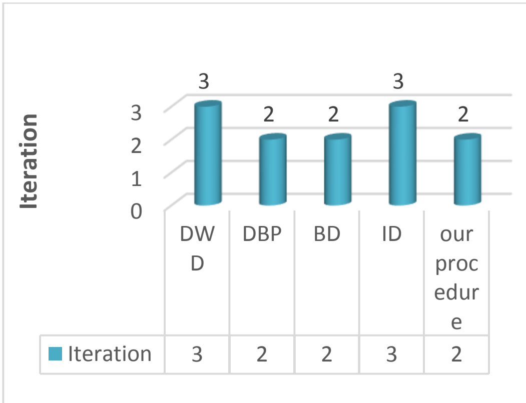

Fig. 3. Iteration Comparison of Decomposition Procedures for Example-2

Fig. 4. Iteration Comparison of Decomposition Procedures for Example-3

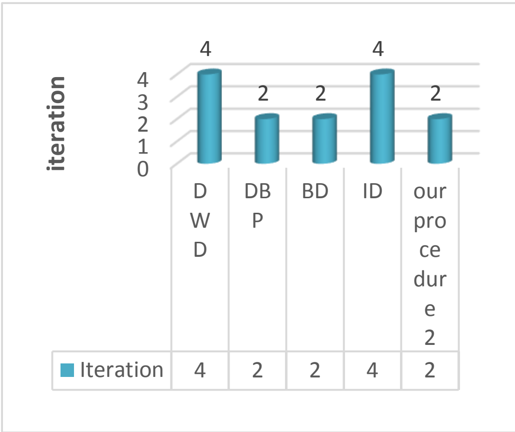

Fig. 5. Iteration Comparison of Decomposition Procedures for Example-4

Fig. 6. Iteration Comparison of Decomposition Procedures for Example-5

CONCLUSION

The heuristic algorithm, proposed to solve IPPs, using the concept of Benders' decomposition and DBP methods. We also generated some conditions for solving problems having either infeasible or unbounded solutions. Upon comparison and rigorous analysis, we observed that our algorithm produced the most efficient computational solutions of IPPs. We also discuss the use of the decomposition methods to develop a heuristic algorithm, describe the limitations of the classical algorithms, and present extensions enabling its application to a broader range of problems.

REFERENCES

Benders J.F. (1962.). Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematik* **4**:238–252.DOI:https://doi.org/10.1007/BF01386316

Cornuéjols Gérard (2008). Valid Inequalities for Mixed Integer Linear Programs. *Mathematical Programming Ser. B* **112**: 3–44. DOI: https:// doi.org /10.1007/s10107-006-0086-0

Cornuéjols Gérard (2007). Revival of the Gomory Cuts in the 1990s. *Annals of Operations Research* **149**: 63–66.

DOI:https://doi.org/10.1007/s10479-006-0100-1

Costa A. M. (2005), A survey on Benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research* **32**: 1429-1450. DOI: https://doi.org/10.1016/j.cor.2003.11.012

Costa A.M.(2018). Improved integer programming models for simple assembly line balancing and related problems. International *Transactions in Operational Research* **25**: 1345–1359. DOI:https://doi.org/10.1111/itor.12206

Crowder H., Johnson EL. & Padberg M. (1983). Solving large scale zero one linear programming problems. *Operation research* **31(5):** 803-834.

DOI: https://doi.org/stable/170888

Dantzig GB & Wolfe P (1960). Decomposition principle for linear programs. *Operations Research* **8(1):** 101-111.DOI: https:// doi.org/10.1287/opre.8.1.101

Das H. K. & Sutradhar Abir (2020). Analyzing Decomposition Procedures in LP and Unraveling for the Two Person Zero Sum

Game and Transportation Problems. *Int. J. of Smart Vehicles and Smart Transportation* **3(1):**21-41. DOI: https://10.4018/IJSVST. 2020010102

Das H.K. & Chakroborty Sajal (2016). Analyzing on the Decomposition based Pricing Procedure for Solving Two Person Zero Sum Game Problems through Computer Algebra. *International Journal of Computer Applications* **156(12**): 37-47. DOI: https://doi.org/10.5120/ ijca.2016912578

Das H.K. & Hasan M. B. (2013). An improved Decomposition Approach and Its Computer Technique for Solving Primal Dual LP & LFP Problems. *Ganit Journal of Bangladesh Mathematical Society* **33**: 65-75. DOI: https: //doi.org/10.3329/ ganit.v33i0. 17649

Das H..K., Islam A. & Hasan M.B. (2014). A new Decomposition-Based Pricing Technique for Solving Large-Scale Mixed IP with a Computer Technique. *Ganit Journal of Bangladesh Mathematical Society* **34:**5-20. DOI: https: //doi.org/10.3329/ganit.v34i0. 28549

Geoffrion M. (1974). Lagrangian Relaxation for Integer Programming. *Mathematical Programming Study* **2**: 82-114.

DOI: https://doi.org/10.1007/BFb0120690

Gomory R.E. (1958). Outline of an algorithm for integer solution to linear programs, Bull. *American Math. Soc*. **64:** 275-278.

Gomory R.E. (1960). Solving linear programming problems in integer, Combinatrial Analysis, Proceedings of Symposia in Applied mathematics X, *American Math. Soc.*, pp. 211-215.

Gomory R.E. (1963). An algorithm for integer solutions to linear programs, *Recent advances in Mathematical programming,* McGraw-hill, New York, USA.

Gupta P. K & Hira D.S. (2005) Problems in Operations Research Principles and Solution, *Chand & Company LTD.*, New Delihi-110055, India.

Hossain M. I. & Hasan M. B. (2013).A Decomposition Technique For Solving Integer Programming Problems. *GANIT J. Bangladesh Math. Soc* **33**: 1 11.

DOI:https://doi.org/10.3329/ganit.v33i0.17649

Hooker JN (1995) Logic-Based Benders Decomposition.

Hussain MA & Bhuiyan MAI (2018). An Improved Decomposition Technique for solving Integer Programing Problems.*Int. Journal of Computer Applications* **182**(23): 0975–8887. DOI: https:// doi.org/10.5120/ijca 2018918023

Mamer JW & McBride RD (2000). A decomposition-based pricing procedure for large-scale linear multicommodity flow problem, *Management Science* **46**(5): 693–709.DOI:https://doi.org/10.1023/B:COAP.0000008652.29295.eb

Marchard H., Martin A, Weismantel R & Wolsey L. (2002). Cutting Planes integer and mixed integer programming, Discrete Apllied Mathematics. **123**(1-3): 387-446. DOI: https:// doi.org/ 10.1016/S0166-218X (01)00348-1

Ravindran, Philips & Solberg (2000). Operations Research. *John Wiley and Sons*, 2nd edition, New York, U.S.A.

Taha HA (2007) .*Operations Research an introduction,* 8th edition, Pearson Princeton hall, New Delhi, India.

Winston Wl (1994). Linear Programming: Applications and Algorithm. *Dunbury press*, Bellmont, California, U.S.A.

Wolsey LA (1998). Integer Programming. *John Wiley & Sons*, Inc. programs: an application to the linear.

(Received 3 January 2020; revised accepted 8 July, 2020)