A REVISIT TO THE GENERALIZED ASSIGNMENT PROBLEM: AN ASCENDING HYPER-PLANE APPROACH THROUGH NETWORK FLOWS

Elias Munapo
Department of Decisions Sciences
University of South Africa
P. O. Box 392, Unisa 0003
Pretoria, South Africa
(munape@unisa.ac.za)

Santosh Kumar
Department of Mathematics and Statistics
University of Melbourne
Parkville, Melbourne, Australia
(skumar@ms.unimelb.edu.au)

and
Senelani D. Musekwa
Department of Applied Mathematics
National University of Science and Technology
PO Box AC 939, Ascot, Bulawayo, Zimbabwe.
(sdmusekwa@nust.ac.zw)

Abstract : This paper presents an ascending hyper-plane approach for solving the generalized assignment problem (GAP). The GAP model is first relaxed to form a transportation model which is easier to handle than the original model. This relaxed model is then formulated as a Minimum-Cost Network Flow Problem (MCNFP) and an efficient network simplex method applied to solve the relaxed problem. The optimal solution of the relaxed model gives a lower bound (LB) to the given GAP. The LB becomes an optimal solution to the GAP, if all resource constraints are satisfied. However, if any resource constraint is not satisfied, that violation is used to determine the new (LB), which is greater than the previous one and hence the ascending hyper-plane approach. These violated resource constraints in the given GAP model and are used to modify the MCNFP diagram before resolving the flow problem. This procedure is repeated until all resource constraints are satisfied in the original GAP model. The proposed method is efficient for the generalized assignment problem.

Keywords : Generalized assignment problem, Transportation model, Minimum-cost-network flows, Network algorithm.

Introduction

The generalized assignment problem (GAP) has many real applications. These include vehicle routing, resource allocation, supply chain, machine scheduling and location. The GAP model, like all other NP hard integer problems, is very difficult to solve using the available linear programming based algorithms. There are several exact approaches [4, 5, 6, 9, 13, 14, 15] for solving the GAP model and their performances have not been satisfactory. Consequently several heuristics [1, 2, 8, 11, 12, 17, 18, 19], which give near optimal solutions have been proposed. In this paper we propose an efficient exact solution method for GAP model. In the proposed approach, the given GAP model is first relaxed into a transportation model which is easier to handle than the original form. This is then formulated as a Minimum-Cost Network Flow Problem (MCNFP) and an efficient network simplex method [16] applied to obtain an optimal solution. The optimal solution is used to determine violated resources constraints in the original GAP model and these violations are used to modify the MCNFP diagram before resolving. The concept used here is similar, but approach is entirely different to that was used in [7]. The process is repeated until all resource

constraints are satisfied. The pivots in the network simplex algorithm involve only simple additions and subtractions and as result the proposed algorithm is very efficient for the GAP algorithm.
This paper has been organized in 7 sections. The mathematical model for the GAP is given in Section 2. Mathematical development of the proposed algorithm has been presented in Section 3. Steps of the algorithm are described in Section 4. An illustrative example is discussed in Section 5. Computational experiments with the proposed algorithm are presented in Section 6 and finally the paper is concluded in Section 7.

The Generalized Assignment Problem

The generalized assignment problem may be formulated as:

$$Z_{GAP} = \text{Minimize } \sum_{i}^{m}\sum_{j}^{n} c_{ij}x_{ij}$$

$$\text{Subject to } \sum_{j}^{n} r_{ij}x_{ij} \le b_i, \forall i$$

$$\sum_{i}^{m} x_{ij} = 1, \forall j$$

$$x_{ij} = 0 \text{ or } 1$$

$$i = 1,2,...m \text{ and } j = 1,2,...n \qquad (1)$$

Where

$i = 1,2,...m,$ is a set of agents, $j = 1,2,...n,$ is a set of tasks, c_{ij} is the cost of assigning agent i to task j and r_{ij} is the resource needed by agent i to do task j. The b_i is the resource available to agent i.

Mathematical development

Relaxed form of the generalized assignment problem

The generalized assignment problem can be relaxed to become an ordinary transportation problem, which is easier to handle than the original GAP model, see Munapo [9].
The restrictive resource constraints in (1) are expressed as (2),

$$\sum_{j}^{n} r_{ij}x_{ij} \le b_i, \forall i \qquad (2)$$

These constraints (2) can be replaced by an inequality given by (3).

$$\sum_{j}^{n} x_{ij} \le \gamma_i, \forall i \qquad (3)$$

Here γ_i are integer values to be determined by solving the knapsack problem (5).
Assuming γ_i are known, the model (1) becomes a transportation problem as presented by (4).

$$\left.\begin{aligned} Z_{relaxed} &= \text{Minimize } \sum_{i}^{m}\sum_{j}^{n} c_{ij}x_{ij} \\ \text{Subject to } &\sum_{j}^{n} x_{ij} \le \gamma_i, \forall i \\ &\sum_{i}^{m} x_{ij} = 1, \forall j \end{aligned}\right\} \qquad (4)$$

Where γ_i, is a constant obtained by solving the knapsack problem given in (5).

$$\left.\begin{aligned} \gamma_i &= \text{Maximize } \sum_{j}^{n} x_{ij} \\ \text{Subject to } &\sum_{j}^{n} r_{ij}x_{ij} \le b_i \\ x_{ij} &= 0 \text{ or } 1 \end{aligned}\right\} \qquad (5)$$

The solution to problem (5) is easy and values of for each i become known in the transportation model (4). The optimal solution to (4) has the property that it gives a lower bound to the GAP model. Thus we have inequality (6).

$$Z_{GAP} \ge Z_{relaxed} \qquad (6)$$

Proof of the inequality (6)

The inequality (6) holds since the constraint $\sum_{j}^{n} r_{ij}x_{ij} \le b_i$ in model (1) has been replaced by the relaxed constraint $\sum_{j}^{n} x_{ij} \le \gamma_i$ in model (4). The assignment constraints, $\sum_{i}^{m} x_{ij} = 1$ remains the same in both models (1) and (4).

For the constraint : $\sum_{j}^{n} r_{ij} x_{ij} \le b_i$, a maximum number (γ_i) exists that can be combined to consume the resource bi. Since coefficients are different, not all combinations will remain feasible. Thus γ_i is the maximum number of jobs that can be assigned to agent i. Therefore the constraint: $\sum_{j}^{n} x_{ij} \le \gamma_i$ is a relaxed one compared to the constraint (2). For a minimization problem the optimal solution of (1) will be greater than or equal to the solution obtained for the relaxed problem (4). However, if the solution to (4) is also feasible to (1), it becomes an optimal solution to the GAP model; otherwise it becomes a lower bound of the required optimal solution to (1). Thus the optimal solution to model in (1) can be obtained by controlled ascending from the optimal solution of the model (4).

3.2 Solution of the knapsack problem

The optimal solution to the knapsack model (5) can be easily obtained by arranging the coefficients in ascending order. i.e. *ascending* $[r_{i1}, r_{i2}, ..., r_{in}] = \{\bar{r}_{i1}, \bar{r}_{i2}, ..., \bar{r}_{in}\}$, where

$$\bar{r}_{i1} \le \bar{r}_{i2} \le ... \le \bar{r}_{in} . \qquad (7)$$

The γ_i in this case is the largest nonnegative integer such that

$$b_i \ge \bar{r}_{i1} + \bar{r}_{i2} + ... + \bar{r}_{i\alpha_i} \qquad (8)$$

In other words γ_i is the largest number of variables whose coefficients can be added in such a way that the resource b_i is not exceeded. This γ_i is now the supply in the relaxed transportation model.

3.3 The transportation model

The transportation problem is shown in Table 1.

Table 1: Transportation problem

	1	2	…	n	**Supply**
1	c_{11}	c_{12}	$\cdots$	c_{1n}	γ_1
2	c_{21}	c_{22}	$\cdots$	c_{2n}	γ_2
…	…	…	…	…	
m	c_{m1}	c_{m2}	$\cdots$	c_{mn}	γ_m
Demand	1	1		1	

The transportation problem presented in Table 1 is not a balanced model. i.e.:

$$\sum_{i}^{m} \gamma_i \ge n \qquad (6)$$

To balance the transportation problem, a dummy column is added and the optimal solution can be efficiently found by using a network simplex method [16]. A balanced transportation problem is shown in Table 2.

Table 2: Balanced transportation problem

	1	2	…	n	Dummy	**Supply**
1	c_{11}	c_{12}	$\cdots$	c_{1n}	0	γ_1
2	c_{21}	c_{22}	$\cdots$	c_{2n}	0	γ_2
…	…	…	…	…	…	…
m	c_{m1}	c_{m2}	$\cdots$	c_{mn}	0	γ_m
Demand	1	1		1	$\sum_{1}^{m}(\gamma_i) - n$	

The optimal solution to the transportation model will act as a lower bound to the generalized assignment problem and is usually infeasible to the original GAP model.

3.4 The transportation problem as an MCNFP

Every balanced transportation problem can be reformulated as a minimum cost network flow problem (MCNFP), see [16]. There are several ways to carry out this representation. In this paper the relaxed transportation model is reformulated as an MCNFP by adding two dummy nodes. A dummy source node (Sso) is introduced on the left and a dummy sink node (Ssi) on the right of the balanced transportation network. The m arcs connecting the source node to the supply nodes have the same costs, (zero in this case) and their flow limits are $\gamma_1, \gamma_2, .., \gamma_m$ respectively. The arcs linking the demand nodes to the sink have the same costs (zeros also) and capacity limits 1,1,…,1, $\sum_{i}^{m} \gamma_i - n$, respectively. The limit $\sum_{i}^{m} \gamma_i - n$ comes from the dummy in Table 2. The reformulation of this transportation model as a MCNFP is given in Figure 1.

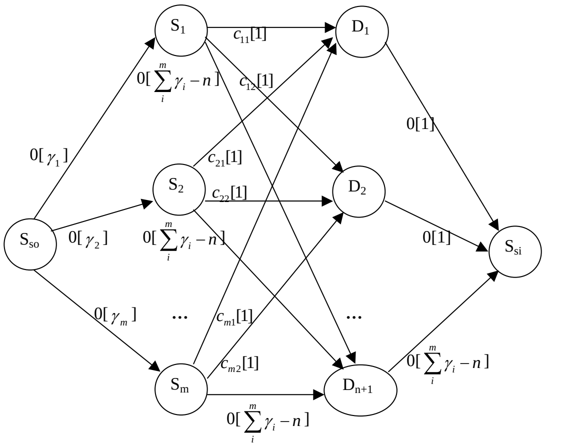

Figure 1: Transportation model formulated as a MCNFP problem

Here the two elements associated with each are x[y] represents that the cost for a unit flow on that arc is x and the flow limit on that arc is y. If the optimal solution to the relaxed problem is also feasible to the resource constraints, it becomes an optimal solution to the GAP model. However, if it is not feasible, some modifications are required, discussed in Section 3.5.

3.5 Modification of MCNFP network when solution of the relaxed problem is infeasible to the given GAP model

The MCNFP network can be modified when the solution of the transportation model is infeasible to the GAP model. The infeasibility arises due to resource constraint (2), as the transportation model is independent of the resource constraints. Suppose a part of an optimal allocation to the MCNFP model contains basic variables for the row i are given as:

$$x_{ij} = ... = x_{ik} = ... = x_{il} = 1 \qquad (7)$$

When this solution is checked for the resource constraint (2) of the original GAP model, they may not be satisfied. For the violated constraints, for example:

$$x_{ij} + ... + x_{ik} + ... + x_{il} \le \tau_r \qquad (8)$$

Where τ_r is the given resource value or some constant. If a total number s of these knapsack constraints are generated before finding an optimal solution then

$$r = 1, 2, ..., s \qquad (9)$$

Only a part of the MCNFP diagram is reconstructed as shown in Figure 3.

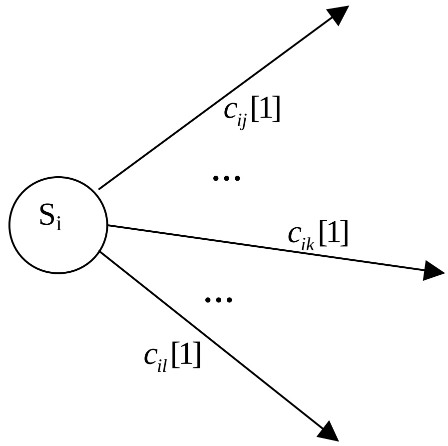

Figure 2: Basic arcs in the MNCFP diagram

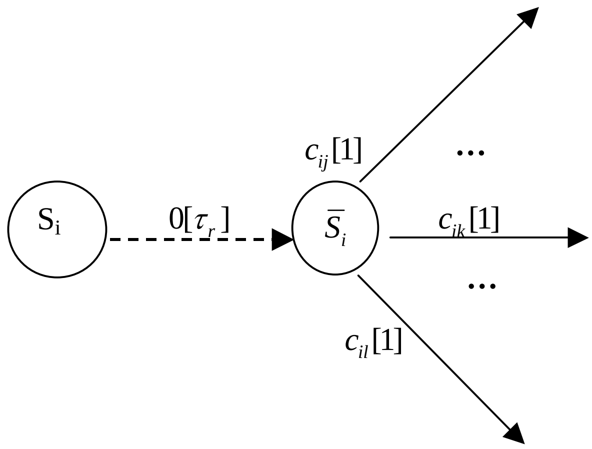

Figure 3: Reconstruction in the MNCFP diagram

Where the arc from node Si to node $\bar{S}_i$ is a dummy arc and node $\bar{S}_i$ is a dummy node. With this approach we only reconstruct the part of the network that is violated. This is easily done by adding a dummy arc and a dummy node. The generated knapsack constraints in a single iteration can be more than one. Thus more than one part of the network can be reconstructed in a single iteration. The approach can result into an algorithm whose flow diagram is given in Figure 4.

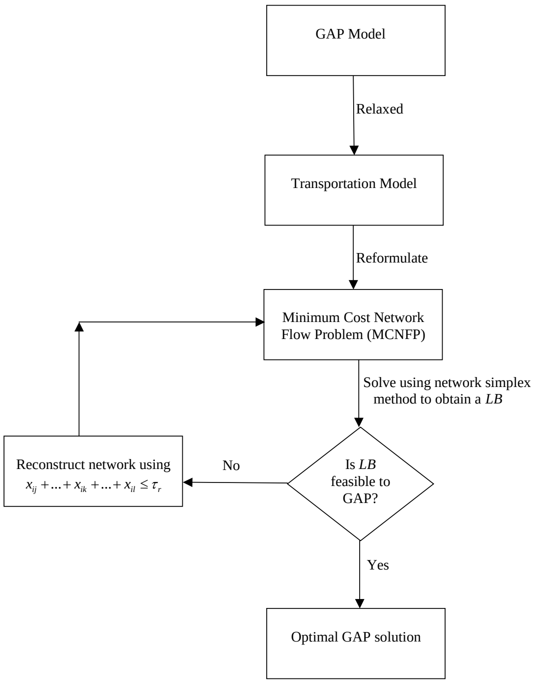

Figure 4: The network algorithm for GAP model

3.6 Optimality and convergence

The reconstruction of the network is exactly the same as adding knapsack constraints to the original generalized assignment problem. These knapsack constraints are generated from the violated GAP constraints and addition of these knapsacks constraints does not change the optimal solution to the GAP model, however, the lower bound gets improved. Addition of knapsack constraints is repeated until the current lower bound satisfies the original GAP model.

$$\left.\begin{aligned} &Z_{LB} = \text{Minimize } \sum_i^m \sum_j^n c_{ij} x_{ij} \\ &\text{Subject to } \sum_i^m x_{ij} = 1, \forall j \\ &x_{ij} + ... + x_{ik} + ... + x_{il} \le \tau_r \\ &x_{ij} = 0 \text{ or } 1 \; \forall ij \end{aligned}\right\} \quad (10)$$

If the lower bound given by the vth iteration is Z_{LBv} then

$$Z_{LB1} < Z_{LB2} < ... < Z_{LBv} < ... < Z_{GAP}$$

The optimal solution to the original GAP model is obtained when Z_{LBv} satisfies (1), i.e.,

$$Z_{LBv} = Z_{GAP}$$

3.7 Infeasibility

If the GAP model is infeasible then the total supply $(\sum_i^m \gamma_i)$ will not be able to reach the sink. The restrictions, $x_{ij} + ... + x_{ik} + ... + x_{il} \le \tau_r$ from the violated constraints will prevent part of the total supply due to additional resources constraints.

4.0 The network simplex algorithm for the GAP

The GAP can be solved as follows:

Step 1: Relax the GAP and form a transportation model. Construct an equivalent MCNFP.

Step 2: Use the network simplex algorithm to find the maximum flow. If the total flow is less than 'n', go to Step 5, otherwise its represents the lower bound (LB) for the GAP. If GAP is satisfied, go to Step 4. Else generate knapsack constraints and go to Step 3.

Step 3: Use the knapsack constraints to reconstruct the network diagram. Return to Step 2.

Step 4: Conclude that optimal solution has been obtained.

Step 5: The GAP has no feasible solution.

5.0 Numerical illustration

Z_{GAP} = Minimize $17x_{11}+19x_{12}+20x_{13}+60x_{14}+50x_{21}+25x_{22}+10x_{23}+23x_{24}$

Subject to:

$$\left.\begin{aligned} &6x_{11}+9x_{12}+6x_{13}+9x_{14}\le 14 \\ &3x_{21}+5x_{22}+9x_{23}+7x_{24}\le 15 \end{aligned}\right\} \quad (11)$$

$$\left.\begin{aligned} &x_{11}+x_{21}=1 \\ &x_{12}+x_{22}=1 \\ &x_{13}+x_{23}=1 \\ &x_{14}+x_{24}=1 \end{aligned}\right\} \quad (12)$$

$$x_{ij} = 0 \text{ or } 1 \; \forall ij \quad (13)$$

Associated transportation problem

Rearrange the coefficients in ascending order in (11), i.e.

$$\begin{aligned} &\{6,6,9,9\} \\ &\{3,5,7,9\} \end{aligned} \quad (12)$$

The values are given by:

$$\left.\begin{array}{l} 6+6=12 \le 14: \ \gamma_1 = 2 \\ 3+5+7 \le 15: \ \gamma_2 = 3 \end{array}\right\} \quad (13)$$

The relaxed transportation model is shown in Table 3 and the corresponding balanced model is given in Table 4.

Table 3: Relaxed transportation model for numerical illustration

	1	2	3	4	**Supply**
1	17	19	20	60	2
2	50	25	10	23	3
Demand	1	1	1	1	

Table 4: Balancing the transportation model

	1	**2**	**3**	**4**	**5(Dummy)**	**Supply**
1	17	19	20	60	0	2
2	50	25	10	23	0	3
Demand	1	1	1	1	(5-4)=1	

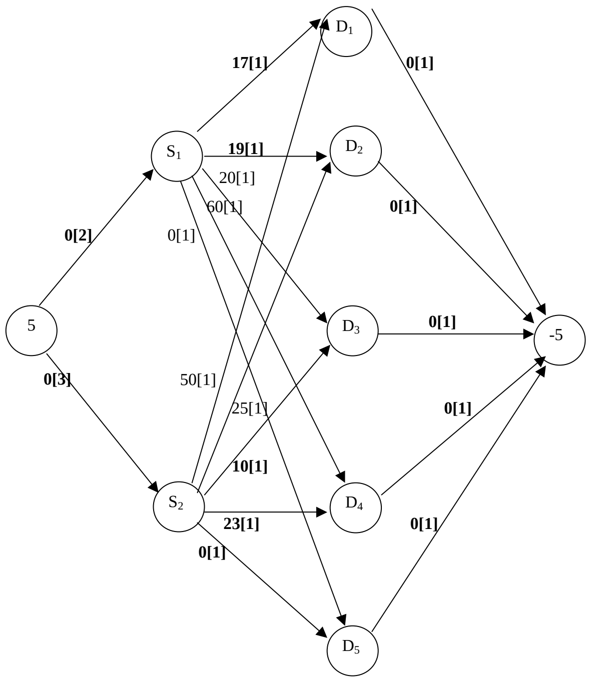

Figure 4: Initial iteration - Network Simplex Method for GAP

Initial lower bound (LB0): $Z_{LB0} = 69 : x_{11} = x_{12} = x_{23} = x_{24} = 1$, x25=1

Real flow is equal to 4, however, this solution is infeasible to the original GAP constraints (11). The knapsack constraints $x_{11} + x_{12} \le 1$ since resource consumed is 15 as against the availability of 14. Similarly the other constraint is $x_{23} + x_{24} \le 1$. The MCNFP model is modified to incorporate these restrictions as shown in Figure 5.

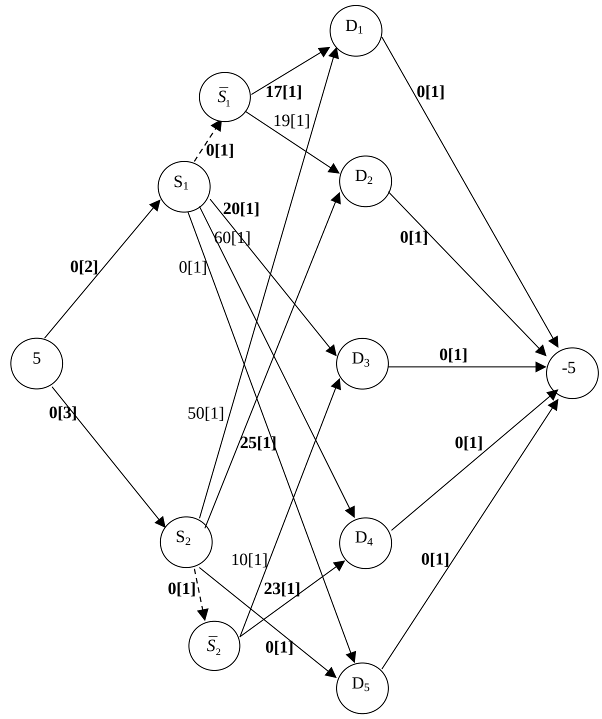

Figure 5: First iteration - Network Simplex Method for GAP

New lower bound (LB1):

$$Z_{LB1} = 85 : x_{11} = x_{13} = x_{22} = x_{24} = 1$$

The new lower bound also satisfies the original GAP constraints given in (11), i.e., $Z_{LB} = Z_{GAP} = 85$ is the required optimal solution. Note that $Z_{LB0} < Z_{LB1}$ =ZGAP.

6.0 Computational results

Instances were generated for four categories of GAP benchmark problems and are denoted by B, C, D and E. The network simplex algorithm proposed in this paper was compared to the stabilized branch and cut and price method [13]. Fortran 2003 [3] was used in coding the two algorithms and running on a DELL OPTIPLEX 760. Computational results were noted as shown in Table 5.

Instances where randomly generated in the ranges given as follows:

6.1 Type B

$$5 \le r_{ij} \le 25; 10 \le c_{ij} \le 50;\ b_i = 0.7\{0.6(n/m)15 + 0.4 \max_i \sum_j i_j^{\min} a_{ij}\}$$

$$i_j^{\min} = \min\{i : c_{ij} \le c_{kj}, \forall k, k = 1,2,...,m\}$$

6.2 Type C

$$5 \le r_{ij} \le 25; 10 \le c_{ij} \le 50 \text{ and } b_i = 0.8\sum_j \frac{a_{ij}}{m}$$

and

6.3 Type D

$$1 \le r_{ij} \le 100; c_{ij} = 111 - r_{ij} + e_{ij};$$

$$b_i = 0.8\sum_j \frac{a_{ij}}{m}; -10 \le e_{ij} \le 10$$

and e_{ij} is also random number.

6.4 Type E

$$r_{ij} = 1 - 10\ln e_{ij}, 0 < e_{ij} \le 1, c_{ij} = 1000/a_{ij} - 10\bar{e}_{ij}; 0 \le \bar{e}_{ij} \le 1,$$

$b_i = 0.8\sum_j \frac{a_{ij}}{m}$ and $\bar{e}_{ij}$ is also random number.

Table 5: Computational results

Problem Type m n Number of binary variables Stabilized branch and cut and price algorithm (seconds) Network algorithm (seconds)

Problem Type	*m*	*n*	Number of binary variables	Stabilized branch and cut and price algorithm (seconds)	Network algorithm (seconds)
B	5	100	500	4.05	3.21
B	10	100	1000	21.92	6.27
B	20	100	2000	29.65	5.34
B	5	200	1000	39.23	7.12
B	10	200	2000	37.88	5.55
C	5	100	500	7.12	4.33
C	10	100	1000	8.93	3.41
C	20	100	2000	17.13	5.46
C	5	200	1000	45.31	8.54
C	10	200	2000	97.12	13.27
D	5	100	500	7.91	3.67
D	10	100	1000	20.64	3.43
D	20	100	2000	88.74	13.58
D	5	200	1000	27.13	5.54
D	10	200	2000	130.48	17.22
E	5	100	500	14.96	4.18
E	10	100	1000	9.61	6.42
E	20	100	2000	89.47	17.71
E	5	200	1000	9.53	4.18
E	10	200	2000	95.78	19.95

The computational results show that the network algorithm proposed in this paper is better than the stabilized branch and cut and price algorithm for solving GAP models.

7.0 Conclusions

It may be concluded that the simplex network algorithm is more efficient than the LP based type of approaches for the GAP model. Reconstructing the MCNFP using the violated constraints is an easy step. The pivots in the network simplex algorithm involve only additions and subtractions. Parallel processors can handle violated constraints independently. Attempts are being made to refine the proposed network algorithm to its full computational efficiency level. The computational results show that it is extremely important to reformulate LPs as MCNFP if at all possible.

References

1. Y. Asahiro, M. Ishibashi and M. Yamashita, Independent and cooperative parallel search methods for the generalized assignment problem, Optimization methods and Software 18(2) (2003), 129-141.
2. P.C. Chu and J.E. Beasley, A genetic algorithm for the generalized assignment problem, Computational Operations Research 24 (1997), 17-23.
3. Fortran 2003 software.
4. M.L. Fisher, R. Jaikumar and L.N. van Wassenhove, A Multiplier Adjustment Method for the Generalized Assignment Problem, Mangement Science 32(9) (1986), 1095-1103.
5. M. Guignard and M. Rosenwein, An Improved Dual-Based Algorithm for the Generalized Assignment Problem, Operations Research 37(4) (1989), 658-663.
6. N. Karabakal, J.C. Bean and J.R. Lohmann, A steepest Descent Multiplier Adjustment Method for the Generalized Assignment Problem. Report 92-11, University of Michigan, Ann Arbor, MI (1992).
7. Kumar, S., Munapo, E and Jones, BC., (2007), An integer equation controlled descending path to a pure integer program, Indian Journal of Mathematics, Vol. 49, No. 2, pp 211-237.
8. M. Laguna, J.P. Kelly, J.L. Conzalez-Velarde and F.F. Glover, Tabu search for the generalized assignment problem. European Journal of Operational Research 82 (1995), 176-189.
9. S. Martello and P. Toth, An algorithm for the generalized Assignment Problem. In Operations Research 81. J.P. Brans (ed.), North-Holland, Amsterdam, (1981), 589-603.
10. E. Munapo, Transportation branch and bound algorithm for the Generalized Assignment Problem, paper communicated for publication (2010).
11. R.M. Nauss, Solving the generalized assignment problem: An optimizing heuristic approach. INFORMS Journal of Computing 15(3) (2003), 249-266.
12. I.H. Osman, Heuristics for the generalized assignment problem: Simulated annealing and tabu search approaches. OR Spektrum 17 (1995), 211-225.

13 A. Pigatti, M. Poggie de Aragao and E. Uchoa, Stabilized branch and cut and price for the generalized assignment problem. In 2nd Brazilian Symposium on Graphs, Algorithms and Combinatorics. Electronic Notes in Discrete Mathematics, Elsevier, Amsterdam, Vol. 19 (2005), 389-395.

14 G.T. Ross and R.M. Soland, A Branch and Bound algorithm for the Generalized Assignment Problem, Mathematical Programming 8 (1975), 91-103.

15 M. Savelsburgh, A branch and price algorithm for the generalized assignment problem, Operations Research 45(6) (1997), 831-841.

16 W.L. Winston, Operations Research: Applications and Algorithms, Duxbury, 4th Edition (2004), 449-465.

17 M. Yagiura, T. Ibaraki and F. Glover, An ejection chain approach for the generalized assignment problem, Informs Journal of Computing 16, (2004), 133-151.

18 M. Yagiura, T. Ibaraki and F. Glover, A path re-linking approach with ejection chains for the generalized assignment problem, European Journal of Operational Research 169, (2006), 548-569.

19 M. Yagiura, T. Yamaguchi and T. Ibaraki, A variable depth search algorithm with branching search for the generalized assignment problem, Optimization Methods and Software 10, (1998), 419-441.