Vol. XII & Issue No. 8 August - 2019
INDUSTRIAL ENGINEERING JOURNAL

DIFFERENT APPROACH FOR SOLVING GENERALIZED ASSIGNMENT PROBLEM

Adlino Vital Afonso
Shridhar Mhalsekar
Do Rosario E Souza Jacqueline

Abstract:

Every organizations objective is to maximize its profit or minimize their costs incurred on the resources. One of the immediate opportunity is in the Assignment problem, which involves assignment of right task to a right agent, that is assigning a job to a machine or to a worker in order to minimize the cost of performing the job on that machine (or by the worker). This is a standardize Assignment problem.When constraints are taken into consideration, that is, when the agents are assumed to have a limited capacity, we have the Generalised assignment problem. Generalised assignment problem is a well-known NP-hard combinatorial problem. The objective is to find the maximum profit or minimum cost of assignment of n jobs to m agents such that each job is assigned to exactly one agent for utilizing the resources offered by the agent but not exceeding its capacity. On the basis of dominant principle technique an Arbitrage method is developed for solving Generalised Assignment Problem. The algorithm used in this method can obtain the optimal solution in minimum computational time. Simulation of the method for solving the generalised assignment model was done using MATLAB. The proposed method is then applied to problems defined as per open source standard OR library available for Generalised Assignment Problem. These solutions were compared with the important standard heuristics. The simulation of generalised assignment problem using MATLAB gives near optimal solutions for small sized problems and an optimal solution for large sized problems defined in standard 0R library.

KEYWORDS: Generalised Assignment Problem, Simulation, Arbitrage, Optimal Solution, Dominant Principle, Heuristics.

1. INTRODUCTION:

Any organization in the world operates with a sole objective to maximize its yearly profits thereby minimizing its costs or expenditures. This can be achieved, if and only if, the tasks performed in an organization are assigned to a right employee or worker (well qualified and vast experience in performing the specific task). It can also be right task performed by right machine and extended to several other real life industrial assignments. The above problem's solution can be achieved by the assignment model, which is a special case of transportation model where in one worker performs only one task, thereby minimizing the overall costs. If the employee or the worker is constrained by its capacity, it becomes a Generalised Assignment Problem.

Here we propose a method called an 'Arbitrage Method for Solving Generalised Assignment Problem'. Arbitrage is a business term which means a practice of buying something in one place and selling it in another place where the price is higher to make profit. The principle on which this method converges to optimality.

The proposed Arbitrage method is a heuristic approach (unlike meta-heuristic) for solving Generalised Assignment Problem (GAP). This method searches for right assignment of jobs to right agents based on highest profit at a low cost while not exceeding the cost capacity of each agent.

2. LITERATURE REVIEW:

The classical assignment problem study goes back in history to the work of G. Monge in late 18th century, although the study was based on linear programming of transportation problem (Demand and capacity equals one). To solve the assignment problem, a Hungarian method was developed by Harold Kuhn (1955). Several other methods for solving the assignment problem are also known.

The Mathematical Formulation of Assignment Model:

Consider an assignment of n-jobs to n-agents. Let C_{ij} be the cost incurred on assigning i^{th} job to j^{th} agent. X_{ij}=1 - agent i is assigned to job j. X_{ij}=0 - Otherwise.

$$Minimize, \quad \sum_{i=1}^{n}\sum_{j=1}^{n} X_{ij}C_{ij} \qquad (1.1)$$

$$S.T \sum_{i=1}^{n} X_{ij} = 1, \qquad j = 1, \dots\dots n \qquad (1.2)$$

$$\sum_{=1}^{n} X_{ij} = 1, \qquad i = 1, \dots\dots n \qquad (1.3)$$

Generalised assignment problem (GAP) is a NP-hard combinatorial problem. It finds the maximum profit or minimum cost assignment of n jobs to m agents such that each job is assigned to exactly one agent for utilizing the resources offered by the agent but not exceeding its capacity, Feltl (2003).

Mathematical formulation of GAP:

Let I be a set of agents and let J be a set of jobs. I=(1,2,.....,b) and J=(1,2,......,a).

We define C_{ij} as the cost (or P_{ij} profit) of assigning job j to agent i.

W_{ij} be the resources (or the weights) required by agent i to perform job j.

b_i be the availability of resources (or capacity) of agent i.

X_{ij}=1 if agent i performs job j.

X_{ij}=0 Othewise.

Therefore we have,

$$Maximise, \quad \sum_{i=1}^{b}\sum_{j=1}^{a} P_{ij}X_{ij} \qquad (1.4)$$

$$S.T \sum_{j=1}^{a} W_{ij}X_{ij} \le b_i, \qquad i \in I = 1,2, \dots\dots, b \qquad (1.5)$$

$$\sum_{i=1}^{b} X_{ij} = 1, \qquad j \in J = 1,2, \ldots \ldots, a \qquad (1.6)$$

$$X_{ij} = 0 \text{ or } 1, \qquad i \in I, j \in J \qquad (1.7)$$

There are various solution methodologies for generalised assignment problem. Among them all, the work by Ross and Soland (1975) was the first on GAP. They proposed a branch and bound method. Later Savelsbergh (1997) introduced branch and price approach and Nauss [2003] soon after followed with his branch and cut approach. In 2006, Nauss (2006) proposed an integer programming methodology for solving GAP, but this took longer computational time to obtain solution for large sized GAP problems. The heuristic approaches for solving GAP were developed in early 1990s. Heuristic algorithms are designed to generate near optimal solutions. Heuristics use linear programming as well as several Lagrangian relaxation methods; Narciso and Lorena (1996), Haddadi and Ouzia (2004), proposing the Lagrangian Relaxation with surrogate relaxation by the former and Integrated Lagrangian Relaxation and Subgradients by later. Monfared (2006) and V. Jeet and E. Kutanoglu (2007) introduced their modifications to Lagrangian Relaxation to obtain their methodologies. S. Raja Balachandar and K.Kannan (2009) proposed a new heuristic approach for solving GAP on based dominant principle.

Various Meta-heuristic search techniques were introduced specially to obtain solution for large sized problems.. Tabu search algorithm using an ejection chain approach was devised by Yagiura et al (2004). Chu and Beasley (1997) gave a genetic algorithm based solution for generalised assignment problem while Amini and Racer (1994) gave a computational comparison of alternative solution methods. Osman (1995) proposed a simulated annealing approach and Feltl (2004) proposed a hybrid genetic algorithm to solve GAP. Cattrysse and Wassenhove (1992) gave a survey of algorithms for earliest existing GAP solution methodologies.

Generalised assignment problem has been modeled for many real life applications. In computer networks, machine loading, facility location, resource scheduling, allocation of memory spaces, designing computer networks, vehicle routing problems and so on.

3. METHODOLOGY

3.1 Arbitrage Methodology:

A unique approach is adapted on the basis of dominant principle technique which searches for the optimal solution based on arbitrage principle. Here, a least or maximum value element as compared to the other elements in the rows or columns of the matrix is considered to be the dominant element. An algorithm is obtained for this approach. This algorithm assumes that the profit to be generated when each job is performed by an agent at a given cost (weights) is known, and in turn does not exceed the overall cost capacity of each agent.

We input all the required data, we initialize matrix X=0 having same order as profit and weight matrix. We find the maximum profit (dominant variable) in each column from the profit matrix and make that corresponding row and column of matrix X_{ij}=1.

We construct matrix Sol by dividing profit matrix by weight matrix. Next we check if x_i£b_i, if not we find the maximum x_j (cost) for that i and equate it to 0, so also Sol_{ij}=0. Now, for that *j* we find maximum Sol_{ij} which means minimum cost, and equate the corresponding value in matrix *X* to 1 (Arbitrage principle). Thus obtaining an updated matrix *X*. Thus satisfying constraint (1.5) after going through a number of iterations. We also check for the constraint (1.6) if satisfied such that the same job is not assigned to two agents.

3.2 Algorithm:

1. *Input data.*
2. *Initialize by assigning 0 to all* X_{ij}.
3. *Calculate, Sol matrix,* $Sol = \frac{P_{ij}}{W_{ij}}$.
4. *Find the max*$(P_{ij})_i$, $\forall j \in J$

   $$X_{ij} = 1$$

5. $x = W.* X$
6. *Check if* $\forall x_i \leq b_i$
7. *if* $x_i > b_i$

   Find max(x_j), $X_{ij} = 0$ *and Sol=0*

   and Find max $(Sol)_j$, $X_{ij} = 1$
8. *Also check if* $\forall j \in J, \sum_{i=1}^{b} X_{ij} = 1$
9. *If* $\forall x_i \leq b_i$ *then, Calculate objective function value* $O = \sum_{i=1}^{b} \sum_{j=1}^{a} P_{ij} X_{ij}$ *else go to step 5.*

4. RESULTS:

The proposed Arbitrage method for solving Generalised Assignment Problem is simulated using MATLAB. The algorithm is tested on 99 test problems ranging from 5 agents/15 jobs to 20 agents/200 jobs to 80 agents/1600 jobs. All these test problems are considered as maximization problems and are taken from open source OR library (www.people.brunel.ac.uk). Of the 99 problems we used, 60 problems are categorised as 'small-sized' problems and 39 problems as 'large-sized' problems. The result of these sets of problems is tabulated in Table 1.

Table 2 compares the results of Arbitrage method for large sized problems with Dominant Principle Heuristics, Dynamic Tabu Tenure with long term memory mechanism and Lagrangian/ Surrogate Relaxation (all four methods consider the problems as maximization problems). Table 3 compares the results of proposed method for problem set 1 to 12 with other existing methods.

These problems are tested on System: Lenovo; Windows 10; Processor: Intel (R) Core (TM) i7-4700MQ CPU @ 2.40GHz, 2401 Mhz, 4 Core (s), 8 Logical Processor(s).

Table 1

SMALL SIZED PROBLEMS						
Problem Set	**No of Agents**	**No. of Jobs**	**Best known Solution**	**Arbitrage Method Solution**	**Time (mSec)**	**Avg. Deviation from Best known(1%)**
GAP 1	5	15	336	**308**	38.509	0.06
GAP 1	5	15	327	**312**	85.528	
GAP 1	5	15	339	**320**	43.537	
GAP 1	5	15	341	**324**	47.597	
GAP 1	5	15	326	**303**	49.144	
GAP 2	5	20	434	**416**	47.053	0.055
GAP 2	5	20	436	**409**	51.741	
GAP 2	5	20	420	**392**	40.695	
GAP 2	5	20	419	**386**	52.914	
GAP 2	5	20	428	**414**	37.532	
GAP 3	5	25	580	**545**	74.279	0.0458
GAP 3	5	25	564	**534**	61.109	
GAP 3	5	25	573	**550**	38.609	
GAP 3	5	25	570	**548**	36.135	
GAP 3	5	25	564	**541**	32.285	
GAP 4	5	30	656	**629**	45.463	0.042
GAP 4	5	30	644	**601**	105.969	
GAP 4	5	30	673	**651**	38.805	
GAP 4	5	30	647	**629**	43.191	
GAP 4	5	30	664	**632**	39.558	
GAP 5	8	24	563	**522**	63.072	0.069
GAP 5	8	24	558	**513**	96.541	
GAP 5	8	24	564	**526**	33.921	
GAP 5	8	24	568	**544**	73.037	
GAP 5	8	24	559	**511**	47.327	
GAP 6	8	32	761	**718**	106.269	0.071
GAP 6	8	32	759	**716**	46.235	
GAP 6	8	32	758	**686**	96.569	
GAP 6	8	32	752	**724**	50.512	
GAP 6	8	32	747	**693**	89.419	
GAP 7	8	40	942	**907**	137.879	0.062
GAP 7	8	40	949	**901**	48.609	
GAP 7	8	40	968	**901**	54.939	
GAP 7	8	40	945	**850**	66.566	

GAP 7	8	40	951	**899**	58.062	
GAP 8	8	48	1133	**1079**	123.44	
GAP 8	8	48	1134	**1063**	66.483	
GAP 8	8	48	1141	**1074**	151.233	0.057
GAP 8	8	48	1117	**1061**	122.309	
GAP 8	8	48	1127	**1048**	144.335	
GAP 9	10	30	709	**640**	40.494	
GAP 9	10	30	717	**673**	46.278	
GAP 9	10	30	712	**660**	84.183	0.079
GAP 9	10	30	723	**675**	50.616	
GAP 9	10	30	706	**634**	137.44	
GAP 10	10	40	958	**933**	139.863	
GAP 10	10	40	963	**885**	120.749	
GAP 10	10	40	960	**901**	148.328	0.0612
GAP 10	10	40	947	**888**	119.266	
GAP 10	10	40	947	**874**	51.475	
GAP 11	10	50	1139	**1077**	64.062	
GAP 11	10	50	1178	**1109**	64.817	
GAP 11	10	50	1195	**1129**	99.144	0.060
GAP 11	10	50	1171	**1109**	148.914	
GAP 11	10	50	1171	**1071**	161.312	
GAP 12	10	60	1451	**1384**	77.464	
GAP 12	10	60	1449	**1396**	70.461	
GAP 12	10	60	1433	**1343**	172.312	0.048
GAP 12	10	60	1447	**1393**	171.511	
GAP 12	10	60	1446	**1356**	137.140	

LARGE SIZED PROBLEMS

Problem Set	No of Agents	No. of Jobs	Best known Solution	Arbitrage Method Solution	Time (mSec)	Avg. Deviation from Best known(1%)
GAP a	5	100	4456	**4448**	50.432	
GAP a	5	200	8788	**8717**	104.941	
GAP a	10	100	4700	**4587**	58.953	0.019
GAP a	10	200	9413	**9265**	155.692	
GAP a	20	100	4857	**4659**	236.893	
GAP a	20	200	9666	**9375**	506.638	

GAP b	5	100	4026	**3833**	416.8	0.039
GAP b	5	200	8502	**8281**	254.445	
GAP b	10	100	4633	**4442**	303.019	
GAP b	10	200	9255	**8963**	278.937	
GAP b	20	100	4817	**4575**	899.810	
GAP b	20	200	9682	**9288**	1193.168	
GAP c	5	100	4411	**4255**	169.359	0.046
GAP c	5	200	8347	**7972**	721.922	
GAP c	10	100	4535	**4262**	115.240	
GAP c	10	200	9258	**8995**	262.843	
GAP c	20	100	4790	**4502**	156.240	
GAP c	20	200	9625	**9147**	1996.121	
GAP d	5	100	9147	**9147**	378.456	0
GAP d	5	200	18750	**18750**	97.605	
GAP d	10	100	10349	**10348**	43.486	
GAP d	10	200	20562	**20562**	49.494	
GAP d	20	100	10839	**10836**	52.380	
GAP d	20	200	21733	**21710**	93.693	
GAP e	5	100	63228	**63228**	35.635	0
GAP e	5	200	128648	**128648**	33.745	
GAP e	10	100	81054	**81054**	34.778	
GAP e	10	200	164317	**164317**	81.111	
GAP e	10	400	316844	**316844**	167.544	
GAP e	20	100	94432	**94432**	838.728	
GAP e	15	900	XX	**789814**	495.887	
GAP e	20	200	XX	**187992**	563.704	
GAP e	20	400	XX	**366771**	607.306	
GAP e	30	900	XX	**878087**	1564.887	
GAP e	40	400	XX	**395832**	1312.197	
GAP e	60	900	XX	**896174**	3723.941	
GAP e	20	1600	XX	**1487990**	3551.759	
GAP e	40	1600	XX	**1580020**	5627.619	
GAP e	80	1600	XX	**1597168**	7593.675	

XX - Data not available

Table 2
Comparison of Large Sized problems with Dominant Principle Heuristics (DPH), Dynamic Tabu Tenure with long term Memory mechanism (TSDl), Lagrangian/ Surrogate Relaxation (RH)

GAP Problem Set	No. of jobs	No. of agents	Optimum/ best solution	**Arbitrage Method**	DPH	TSDL	RH	DPH Solution Time (sec)	AM Solution Time (sec)
A 1	5	100	4456	**4448**	4456	4456	4456	1.36	0.05
A 2	5	200	8788	**8717**	8788	8788	8788	1.74	0.1
A 3	10	100	4700	**4587**	4700	4700	4700	2.52	0.06
A 4	10	200	9413	**9265**	9413	9413	9413	2.02	0.15
A 5	20	100	4857	**4659**	4857	4857	4857	1.97	0.23
A 6	20	200	9666	**9375**	9666	9666	9666	2.86	0.5
B 1	5	100	4026	**3833**	4026	4026	4008	1.69	0.41
B 2	5	200	8502	**8281**	8502	8505	8502	1.19	0.25
B 3	10	100	4633	**4442**	4633	4633	4633	1.94	0.3
B 4	10	200	9255	**8963**	9255	9255	9255	2.33	0.27
B 5	20	100	4817	**4575**	4817	4817	4817	2.53	0.89
B 6	20	200	9682	**9288**	9682	9682	9670	2.76	1.19
C 1	5	100	4411	**4255**	4389	4411	4411	1.79	0.17
C 2	5	200	8347	**7972**	8346	8346	8347	1.46	0.72
C 3	10	100	4535	**4262**	4535	4535	4528	2.12	0.11
C 4	10	200	9258	**8995**	9258	9258	9247	1.9	0.26
C 5	20	100	4790	**4502**	4790	4790	4784	1.94	0.15
C 6	20	200	9625	**9147**	9625	9625	9611	2.89	1.99
D 1	5	100	9147	**9147**	9147	9147	9147	1.63	0.37
D 2	5	200	18750	**18750**	18750	18750	18750	1.83	0.09
D 3	10	100	10349	**10348**	10349	10349	10349	2.32	0.04
D 4	10	200	20562	**20562**	20562	20562	20562	2.32	0.05
D 5	20	100	10839	**10836**	10839	10839	10839	2.43	0.05
D 6	20	200	21733	**21710**	21733	21733	21733	2.77	0.09
E 1	5	100	63228	**63228**	63228	XX	XX	1.55	0.03
E 2	5	200	128648	**128648**	128648	XX	XX	1.73	0.03
E 3	10	100	81054	**81054**	81054	XX	XX	2.21	0.03
E 4	10	200	164317	**164317**	164317	XX	XX	2.46	0.08
E 5	10	400	316844	**316844**	316844	XX	XX	2.57	0.16
E 6	20	100	94432	**94432**	94432	XX	XX	2.45	0.84
E 7	15	900	XX	**789814**	XX	XX	XX	XX	0.49
E 8	20	200	XX	**187992**	XX	XX	XX	XX	0.56
E 9	20	400	XX	**366771**	XX	XX	XX	XX	0.6
E 10	30	900	XX	**878087**	XX	XX	XX	XX	1.56
E 11	40	400	XX	**395832**	XX	XX	XX	XX	1.31
E 12	60	900	XX	**896174**	XX	XX	XX	XX	3.72
E 13	20	1600	XX	**1487990**	XX	XX	XX	XX	3.55
E 14	40	1600	XX	**1580020**	XX	XX	XX	XX	5.62
E 15	80	1600	XX	**1597168**	XX	XX	XX	XX	7.6

XXData not available

Table 3
Comparison of Percentage deviation from best known solution of various methods for solving GAP

Prob set	AM	MTH	FJVBB	FSA	MTBB	SPH	LTIFA	RSSA	TS6	TSI	GA~	GAb
gapl	**6**	5.43	0.13	0	0	0.08	1.74	0	0	0	0	0
gap2	**5.5**	5.02	0	0.19	0	0.11	0.89	0	0.24	0.1	0.01	0
gap3	**4.5**	2.14	0	0	0	0.09	1.26	0	0.03	0	0.01	0
gap4	**4.2**	2.35	0.83	0.06	0.18	0.04	0.72	0	0.03	0.03	0.03	0
gap5	**6.9**	2.63	0.07	0. I 1	0	0.35	1.42	0	0.04	0	0.1	0
gap6	**7.1**	1.67	0.58	0.85	0.52	0.15	0.82	0.05	0	0.03	0.08	0.01
gap7	**6.2**	2.02	1.58	0.99	1.32	0.13	1.22	0.02	0.02	0	0.08	0
gap8	**5.7**	2.45	2.48	0.41	1.32	0.23	1.13	0.1	0.14	0.09	0.33	0.05
gap9	**7.9**	2.18	0.61	1.46	1.06	0.12	1.48	0.08	0.06	0.06	0.17	0
gapl0	**6.1**	1.75	1.29	1.72	1.15	0.25	1.19	0.14	0.15	0.08	0.27	0.04
gapll	**6**	1.78	1.32	1.1	2.01	0	1.17	0.05	0.02	0.02	0.2	0
gapl2	**4.8**	1.37	1.37	1.68	1.55	0.1	0.81	0.11	0.07	0.04	0.17	0.01
Avg % deviation	**5.9**	2.56	0.84	0.72	0.78	0.13	I. 15	0.04	0.06	0.03	0.02	0

MTH: Martello and Toth constructive heuristic;

FSA: Cattrysse fixing simulated annealing algorithm:

SPH: Cattrysse.Wassenhove set partitioning heuristic;

RSSA: Osman hybrid simulated annealing/tabo search;

TS6: Osman long term tabu search with first-admissible selection:

TSI: Osman long term tabu search with best-admissible selection:

GA~: GA without the heuristic operator;.

GAb: GA with the heuristic operator. branch-and-Salomon and Van

AM: Arbitrage Method

5. DISCUSSION:

The results from Table 1 shows 12 sets of problems labeled as small size. The number of Agents ranging from 5 to 10 and the number of jobs ranging from 15 to 60. Each set has 5 problems each. For the ones labeled as large-sized, there are 5 sets of problems namely GAP a, GAP b, GAP c, GAP d, GAP e. Number of agents ranging from 5 to 80 while the number of jobs ranging from 100 to 1600. GAP e having 15 problems and GAP a, GAP b, GAP c, GAP d having 6 problems each.

The proposed Arbitrage method gives a near optimal solution (best known), having a maximum average percentage deviation of 7.9%. For large sized problems where the results were not available, this method could give immediate results with least computational time, as compared to the other methods. This method computed results for all the large sized problems. GAP d and GAP e sets are having 0% deviation from the best optimum known. Thus illustrating the reliable application of the proposed methodology for solving large-sized problems efficiently. The results of large sized problems are compared with three other heuristic approaches, Dominant Principle Heuristics, Dynamic Tabu Tenure with long term memory mechanism and Lagrangian/ Surrogate Relaxation in table 2. It is clear from table 2 and table 3, the Arbitrage method gives improved optimal solution for large-sized problems as compared to small-sized problems.

6. CONCLUSION:

The Arbitrage method for solving Generalised assignment problem is a different approach for solving GAP. It is an exact method which tries to converge to optimality by using minimum resources while maximizing profit within its resource capacity. As the term arbitrage is described, it tries to buy something (or do a job) at one place at lower cost while selling something at another place which is at higher cost in order to satisfy all GAP constraints.

The proposed method is solved for 60 small sized problems and 39 large sized problems taken from the open source OR library (www.people.brunel.ac.uk). The results of these problems are compared with other existing solution methodologies. The results of problem sets, GAP 1 to GAP 12, shows a higher average percentage deviation as compared to the other methods with a maximum of 7.9%. This percentage deviation, as observed in the comparative study of problem sets from GAP a to GAP e for large sized problems is reduced to zero.

Thus it can be concluded, that this method is an heuristic approach for solving Generalised Assignment Problem that gives a near optimal solution. Arbitrage method tries to converge to optimality by searching a reduced search space thereby minimizing the computation time. This method gives improved optimal solutions for large sized problems which makes it reliable to use for such problems. The proposed method is simulated using mathworks MATLAB.

All the problems sets have been considered as maximization problems. A slight modification in the algorithm can make its solution optimal for both maximization as well as minimization problems. Applying this algorithm to one of the GAP variant/ extension with a minimal modification, in regards with a real life application and verifying the results, both these, remain as the future challenges.

REFERENCES:

1. D. G. Cattrysse, L.N.V. Wassenhove, Asurvey of algorithms for the generalized assignment problem. European Journal of Operational Research 60,pp.260-272, 1992.
2. P. C. Chu and J. E. Beasley, A genetic algorithm for the generalized assignment problem. Comput. Oper. Res. 24, pp.17–23, 1997.
3. H. Feltl, An Improved Hybrid Genetic Algorithm for the Generalized Assignment. PhD thesis, Vienna University of Technology, Austria, 2003.
4. H. Feltl and G. Raidl, An improved hybrid genetic algorithms for the generalized assignment problem, SAC '04, Nicosia, Cyprus, march pp.14-17, 2004.
5. M. Galati, Decomposition in Integer Linear Programming, Ph.D. thesis, Lehigh University, 2009.
6. S. Haddadi, H.Ouzia, An effective Lagrangian heuristic for the generalized assignment problem,. INFOR 39, pp.351-356, 2001.
7. S. Haddadi, H.Ouzia, Effective algorithm and heuristic for the generalized assignment problem. European Journal of Operational Research 153, pp.184-190, 2004.
8. Harold W. Kuhn, (2012). A tale of three eras: The discovery and rediscovery of the Hungarian Method.Eur. J. Oper. Res., Vol 219, Issue 3, 16 June 2012, pp. 641-651, 2012.
9. V. Jeet, E. Kutanoglu, Lagrangian relaxation guided problem space search heuristics for generalized assignment problems, European Journal of Operational Research 182, pp.1039-1056, 2007.
10. L. A. N. Lorena and M. G. Narciso, Relaxation heuristics for a generalized assignment problem. Eur. J. Oper. Res. 91, pp.600–610, 1996.
11. S. Martello and P. Toth, An algorithm for the generalized assignment problem. Proc. Ninth IFORS Internat. Conf. on Operational Res.. Hamburg, Germany. pp.589–603, 1981.
12. M. A. S. Monfared. and M. Etemadi., The impact of energy function structure on solving generalized assignment problem using Hopfield neural network, European Journal of Operational Research 168, pp.645-654, 2006.
13. R. M. Nauss, Solving the generalized assignment problem: An optimizing and heuristic approach. INFORMS Journal of Computing 15 (3), pp.249-266, 2003.
14. R. M. Nauss, The generalized assignment problem. In: Karlof, J.K. (Ed.), Integer Programming: Theory and Practice. CRC Press, Boca Raton, FL, pp.39-55, 2006.
15. I. H. Osman, Heuristics for the generalized assignment problem: Simulated annealing and tabu search approaches. OR Spektrum 17, pp.211-225,1995.
16. M. Racer and M. M. Amini,. A robust heuristic for the generalized assignment problem. Ann. Oper. Res. 50 pp.487–503,1994.
17. G. T. Ross and R. M. Soland, A Branch and Bound Approach for the generalized assignment problem, Mathematical Programming, 8(1), pp.91-105,1975.
18. S. Raja Balachandar and K.Kannan,World Academy of Science, Engineering and Technology, International Journal of Mathematical and Computational Sciences, Vol:3, No:11, pp.969-974, 2009.
19. Salim Haddadi and HaceneOuzia (2004). Effective algorithm and heuristic for the generalized assignment problem. Eur. J. Oper. Res. , Vol 153, Issue 1, pp.184-190, 2004.
20. M. Savelsbergh, A branch-and-price algorithm for the generalized assignment problem. Operations Research 45 (6), pp.831-841,1997.
21. M. Yagiura, T. Ibaraki, and F. Glover, An ejection chain approach for the generalized assignment problem, INFORMS Journal on Computing, 2, pp.16-28, 2004.
22. T. Koch, T. Achterberg, E. Andersen, O. Bastert, T. Berthold, R. E. Bixby, E. Danna, G. Gamrath, A. M. Gleixner, S. Heinz, A. Lodi, H. Mittelmann, T. Ralphs, D. Salvagnin, D. E. Steffy, and K. Wolter (2011). MIPLIB 2010, Mathematical Programming Computation, 3, pp. 103–163.

AUTHORS

Adlino Vital Afonso, Goa College of Engineering, Goa, India. Email id : *Adlino_afonso@yahoo.co.in*

Dr Shridhar Mhalsekar, Goa College of Engineering, Goa, India.

Do Rosario E Souza Jacqueline, Rosary College of Arts and Commerce, Goa, India