A Comparison of Exact and Heuristic Methods for a Facility Location Problem

Simone Argenziano [1], Maurizio Naldi [2, 3], Gaia Nicosia [1], Andrea Pacifici [2]

[1] *Dipartimento di Ingegneria*,
Università degli Studi "Roma Tre",
Via della Vasca Navale 79, 00146 Roma, Italy
email: gaia.nicosia@uniroma3.it.

[2] *Dipartimento di Ingegneria Civile e Ingegneria Informatica,*
Università degli Studi di Roma Tor Vergata,
Via del Politecnico 1, 00133 Roma, Italy
email: {maurizio.naldi,andrea.pacifici}@uniroma2.it.

[3] *Dipartimento di Giurisprudenza, Economia, Politica e Lingue Moderne,*
Libera Università Maria SS. Assunta (LUMSA),
Via Marcantonio Colonna 19, 00192 Roma, Italy
email: m.naldi@lumsa.it

Abstract **- We formulate a facility location problem where the demand of any single client must be allocated to a single facility and a prize is obtained by allocating the demand of a client to a certain facility, i.e. a prize-based variant of the Single Source Capacitated Facility Location Problem. For this problem we pursue both an exact approach through Integer Linear Programming and a heuristic approach based on a local search algorithm. We compare both approaches by considering 500+ instances. The heuristic approach allows to obtain a reduction of the computational time by a factor larger than 10 in 92% of instances and 100 in 64% of instances. The time reduction is obtained with a small sacrifice in the value of the objective function that is achieved, smaller than 10% in nearly 70% of cases.**

Keywords - Facility location; Prize collecting; Strategic planning; Emergency logistics.

I. INTRODUCTION

Facility location problems, where a set of items have to be assigned to a number of facilities, represent a well known class of problems in operations research [8], [21], [1]. In the classical facility location problem, we are given a set of facilities, with associated opening costs, and a set of clients. The goal is to open a subset of facilities and to connect each client to the closest open facility, so that the total connection and opening cost is minimized. A special case is the Single Source Capacitated Facility Location Problem (SSCFLP), where the demand of any single client has to be allocated to a single facility and each facility has a limited capacity. Several efforts have been devoted to solving SSCFLP, based either on Lagrangean relaxation [14], [22], [4], [23], [5] or on an exact approach, e.g. branch-and-bound in [12], [12], [11], and branch-and-cut in [16].

Hereafter we study a variant of this locational decision, in which a prize is associated to allocating a client's demand to a certain (open) facility and the objective is the maximization of the overall revenue obtained as the difference between the total allocation prize and the facility-opening costs. For such a problem we introduce the abbreviation PCFL. Differently from the standard facility location, in our variant it is feasible to allocate the demand of a proper subset of clients. A similar problem has been tackled in [15], where the facilities are connected through a Steiner tree, though.

The original motivation for investigating this matter comes from an application related to the emergency caused by the earthquake that hit central Italy in the summer of 2016 (and until 2017), where the construction of prefabricated cottages was among the measures taken to support the affected population in the transition from emergency response to reconstruction. This process involved taking several decisions concerning, in particular, the location of cottages and their assignment to eligible users under budget and time constraints. A list of the potential locations for (groups of) cottages had to be considered, together with their opening cost (including construction and infrastructures setup) and their capacity (basically the amount of people that could be accommodated), taking into account as well the needs expressed by the affected population, represented by a suitable number of points (called clients) each indicative of a portion of the territory and associated to a number (demand) proportional to the amount of population and intensity of the suffered damages. Similar problems arise in many cases of emergency relief operations [3], [2], [24]. Fairness issues also play an important role in this context (as dealt with, e.g., in [7], [18]), as the level of damages suffered by the client and the geographical distance between the facility and the client have also to be considered.

Here we explore the solution of the PCFL problem by adopting both an exact Integer Linear Programming (ILP) approach (in [19] a mixed integer programming model was employed for the optimal allocation of products demand to distribution centres spread over the Italian territory) and a heuristic approach.

After providing a mathematical formulation of the problem in Section II, we describe both approaches in Sections III and IV. In Section V, we show that:

- the heuristic approach may provide more than a tenfold (in 92% of instances) or even more than hundredfold (in 64% of instances) reduction of the CPU time in most cases;
- in all evaluation instances classes but one the objective function value achieved by the heuristic approach has been within 20% of that obtained through the exact approach (with the gap reducing to less than 10% in nearly 70% of instances)

II. Problem Statement

In this section, we provide a formal statement of the PCFL problem.

A set of clients $D = \{1, \ldots, m\}$ require using a certain resource (possibly representing a commodity, a service, or a combination thereof). For each $j \in D$, a demand $d_j \in \mathbb{R}_+$ expressing the amount of resource by client j is given. Resource suppliers are to be selected among a set $S = \{1, \ldots, n\}$ of potential facilities. For each facility $i \in S$, the following quantities are known.

1) The opening cost f_i is paid if the facility i is active (open) and hence clients demand can be allocated to this facility.
2) The capacity k_i is the available amount of resource at facility i.
3) The prizes $p_{i,j} \in \mathbb{R}_+$, for all $j \in D$, are earned if the demand of client j is allocated to facility i.

A set $T \subseteq S$ of active facilities, a set $E \subseteq D$ of served clients, and an allocation $\psi : E \longrightarrow T$ of the client demands to active facilities are sought such that: the overall demand $\sum_{j:\psi(j)=i} d_j$ allocated to any facility $i \in T$, does not exceed its capacity k_i, and the revenue $\sum_{j \in E} p_{\psi(j),j} - \sum_{i \in T} f_i$ is maximized.

From a computational complexity point of view, we note that, even if the set of active facilities T were given, finding an optimal demand allocation ψ is a hard task. A simple reduction from the well known 3-Partition problem [10] shows that the simple allocation decision of the clients demands - even with constant $p_{i,j} = 1$ - to a given set T of capacitated facilities is already a strongly $\mathcal{NP}$-hard problem. In view of this result, it is very unlikely that an efficient (i.e., with a computational cost bounded by a polynomial in m and n) solution algorithm exists for our facility location problem.

III. ILP Model

In this section, we present the mathematical program modeling our locational decision problem. (In the remainder of the paper, whenever this does not generate confusion, we write p_{ij} instead of $p_{i,j}$).

The two sets of variables $x \in \{0,1\}^{|S|}$ and $y \in \{0,1\}^{|S \times D|}$ model the facility location and the client-demand allocation decisions, respectively. The meaning of these quantities is illustrated hereafter.

$$x_i = \begin{cases} 1 & \text{if facility } i \text{ is open} \\ 0 & \text{else} \end{cases} \tag{1}$$

$$y_{ij} = \begin{cases} 1 & \text{if client } j \text{ is served by facility } i \\ 0 & \text{else} \end{cases} \tag{2}$$

Note that, since y_{ij} are binary variables, the demand of any client j can only be *entirely* allocated to a *single* facility. (dubbed as "Single-Source" allocation).

$$\max \sum_{i \in S} \sum_{j \in D} p_{ij} y_{ij} - \sum_{i \in S} f_i x_i \tag{3}$$

$$\sum_{i \in S} y_{ij} \leq 1 \qquad j \in D \tag{4}$$

$$y_{ij} \leq x_i \qquad (i,j) \in S \times D \tag{5}$$

$$\sum_{j \in D} d_j y_{ij} \leq k_i x_i \qquad i \in S \tag{6}$$

$$x_i, y_{ij} \in \{0,1\} \qquad i \in S,\ j \in D. \tag{7}$$

As opposed to classical facility location, Equation (3) is a maximization objective function, comprised of the total prize for serving the demand of some clients net of the facility-opening costs.

Taking into account the integrality of the variables (Equations (7)), the constraints ensure that:

- a client demand may be entirely satisfied by one facility *or* not satisfied at all, as per Equations (4);
- a client can only be served by an active facility, as per Equations (5);
- the total demand allocated to an active facility cannot exceed the capacity of that facility, as per Equations (6).

IV. Local Search algorithms

Similar to what was proposed in [15], we implemented a Variable Neighborhood Local Search heuristic algorithm.

Initial solution. The initial solution is built as follows.

We first sort all pairs $(i,j) \in S \times D$ by non-increasing values of the p_{ij}/d_j ratio. For each pair (i,j) in the given order, if there is still capacity to allocate the demand of j, we assign j to i, update the residual capacity of i, and remove all $(*,j)$ pairs from the list. Eventually we output the set T of active facilities.

Local Search Phase. Given the set of active facilities T, we consider two different types of neighborhood for obtaining new facility sets.

- Facility Insertion/Removal Neighborhood: It contains all the solutions obtained from all possible sets T' such

DOI 10.5013/IJSSST.a.20.S1.17 ISSN: 1473-804x online, 1473-8031 print

that $0.8 \leq \sum_{T'} f_i / \sum_T f_i \leq 1.2$: i.e., the total opening cost of T' differs from that of T of at most 20%.
- Facility Swap Neighborhood: It contains all the solutions obtained by swapping facilities, i.e. replacing one open facility in T by an inactive one in $S \setminus T$ (giving priority to the inactive facilities with larger k_i/f_i).

Once the set of open facilities T is set, the actual assignment of clients to facilities can be performed in several ways. We consider three different approaches, resulting in three different variants of the local search algorithm.

1) *Opt-Assign.* Use ILP to obtain the optimal clients-to-facilities assignment. (We set $x_i = 1$ for all $i \in T$ and solve ILP (3)–(7)).
2) *Heur-Assign.* We use a greedy approach similar to the one used for the initial solution computation.
3) *Mix-Assign.* This approach is a combination of the above two. A given fraction of the clients—the most profitable in terms of the p_{ij}/d_j ratio—is assigned using the above greedy approach (Heur-Assign). The remaining clients are then optimally assigned through the usual ILP.

We have implemented all three versions of the local search algorithm. The results of the computational experiments are reported in the next section.

V. Computational Experiments

The experiments performed to assess the quality of our ILP models has been built starting from the instances of Beasley [6] and Holmberg [13] for SSCFLP. These two test sets are considered among the hardest available for that problem. Each of those instances consists of the following data:

1) $|S| = n$, $|D| = m$ numbers of potential facilities an clients;
2) opening costs $f \in \mathbb{Z}^n_+$ and capacities $f \in \mathbb{Z}^n_+$ of the facilities;
3) demands $d \in \mathbb{Z}^m_+$ of the clients;
4) allocation *costs* $c_{ij} \in \mathbb{Z}$ for all facility-client pairs.

In order to derive the corresponding instance of PCFL, by considering the allocation costs as prizes (i.e., $p_{ij} := c_{ij}$), we leave all the data unchanged but modify the opening costs by a suitable factor h, so that we set $f_j := hf_j$ The purpose is to avoid—with high probability—trivial solutions in which all facilities are active and/or all clients are served. This factor is obtained on the grounds of the following simple considerations.

The average achievable prize if all the clients are served can be computed as $\xi = \sum_S \sum_D p_{ij}/n$. An average number of active facilities needed to satisfy the total clients demand is $\nu = \frac{\sum_D d_j}{\sum_S k_i/n}$. So that an average opening cost for those r facilities can be estimated as $\phi = \nu \sum_S f_i/n$.

We design the new instances in such a way that ξ is substantially larger than ϕ, and hence it would not be profitable to open all the facilities and satisfy all the demand. For instance, if we require $\phi = \ell\xi$, it is straightforward to see that the following factor would do:

$$h = \frac{\ell(\sum_S k_i)(\sum_{S,D} p_{ij})}{n(\sum_S f_i)(\sum_D d_j)}. \tag{8}$$

All the experiments were run on a PC equipped with a Intel core i7 processor (1.80 GHz CPU and 8GB RAM) under Windows 10, 64 bit. All the ILP models (including the assignment for the heuristic algorithms), have been solved by the commercial solver Gurobi™, version 7.5.2.

A first set of experiments was performed by testing our ILP model on 95 instances derived from the above cited test set by Holmberg, and setting $\ell = 2$ for the factor of Equation (8). For all those instances, but in two cases, our integer program and Gurobi were effective and found an optimal solutions around 0.1 seconds. Only for two instances, with $n = 30$ and $m = 200$, it took more that 100 seconds to find an optimum. Starting from the characteristics of these two harder instances, a new set of experiments has been generated. In Table I, the ranges of the parameters defining the new test set are illustrated. For each of the 52 classes, 10 instances have been (pseudo-)randomly generated.

Hereafter, for brevity, we report only the results concerning the ILP model and two versions of the local search algorithm, namely Heur-Assign and Mix-Assign in which the fraction of clients greedily assigned is $\alpha = 75\%$. For all heuristic algorithms a time limit of one hour was set. The results of the Opt-Assign strategy are not reported mainly because of its poor performance on the test instances of Table I. This heuristic is able to solve some instances in the Classes from 1 to 16; however no instances in the other classes have been solved within the time limit. A similar performance is reached by the Mix-Assign local search algorithm when $\alpha = 25\%$ or 50%.

The picture concerning $\alpha = 95\%$ or 90% is different. Both versions of the algorithm have a performance very similar to that of Mix-Assign with $\alpha = 75\%$. In fact, on some instances they run slightly faster and sometimes get slightly worse objective function values.

In Table II we report the results on the objective function value for the ILP, the Heur-Assign and the Mix-Assign heuristics. All the values are mean values computed only over the instances solved to optimality for the ILP and within the time limit for the heuristic. For each of the two reported heuristic, also a measure of the percentage error with respect to the optimum has been evaluated. The column Gap% reports the percentage error $\frac{OPT-Heur}{OPT} \cdot 100$. This value has been computed only for the instances in which both the heuristic and the ILP reached a solution. The symbol "–" in the table means that in all instances of the class Gurobi out of memory (and hence no comparison with the heuristic can be performed).

The asterisk in the Gap column, indicates that no gap computation was done, since the heuristic algorithm returned at least one negative objective function value.

As to the ILP performance, in the majority of the instance classes (29 out of 52) at least one instance in the class ran out of memory.

In Table III the average CPU times (expressed in seconds) required by the different algorithms are reported. In the table, "t.l." indicates that no solution has been found within the time limit of 1 hour.

By looking at Table III, we observe that in most instances the Heur-Assign is the fastest algorithm, the only exception being instance classes 7 and 14 (the ILP works quite well while both heuristics reach the time limit of one hour). Also, comparing the solution value of Heur-Assign and Mix- Assign with _ = 75%, we observe that many time they obtain the same values hence making Heur-Assign the better performing algorithm between the two.

We can get an overall view of the performance of the heuristic approach by defining two relative performance indices: the Time Index and the *Objective Function Index.*

The Time Index (TI) is defined as the ratio of the time required by the ILP approach to that observed with the heuristic approach. The higher the TI, the faster the heuristic approach. We expect the TI to be higher than 1 in nearly all cases, and we wish it to be much larger than 1.

The Objective Function Index (OFI) is instead the ratio of the objective function value achieved by the heuristic approach and that obtained with ILP. We can simply relate it to the Gap index defined early, since the OFI is simply the complement to 1 of the Gap. For each single instance we always have OFI _ 1 (the values being occasionally larger than 1 in Table II are due to the ILP not solving some of the instances in the class). The closer to 1 the OFI is, the better the heuristic approach is.

If we now plot the OFI vs TI for each instance class, we get a scatterplot that takes into account both performance facets. The dots (instance classes) lying closer to the origin represent the cases where the heuristic approach performs worst, while the dots located in the right upper corner represent the cases where the heuristic approach performs best. The resulting scatterplot is shown in Fig. 1. The reduction in the computational time obtained thanks to the heuristic approach can be extremely large. We achieve a reduction larger than one hundredfold in 64.1% of the instances and larger than tenfold in a striking 92.3% of the instances. On the other hand, the sacrifice in the optimization performance (i.e., the objective function value) is quite limited, the gap is lower than 10% in 69.2% of the instances and it is lower than 20% in practically all instances examined (with a single exception).

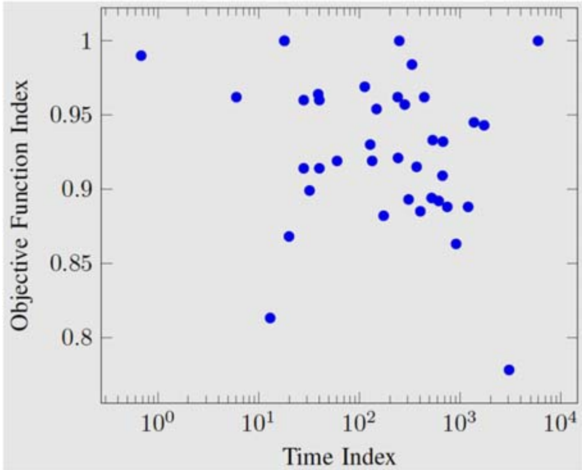

Fig. 1: Performance trade-off

VI. CONCLUSION

The performance of the heuristic approach appear promising. It allows to achieve significant reductions in the computational time (more than one hundredfold in most cases) with a very limited sacrifice in optimization performance (the gap with respect to the ILP is lower than 10% in most cases). In order to obtain a more complete analysis we aim at examining a wider set of instances and arriving at a better description of the trade-off between optimization objective and computational effort. For the same problem we also plan to explore alternative optimization approaches, in particular exact methods like (combinatorial) branch-and-bound that would require designing both suitable branching scheme and developing good quality lower bound (similar to what has been done in, e.g., [9]). Another interesting direction may consider taking a game theoretic approach, where agents (the facilities) compete to get clients, following in the step of several papers where combinatorial optimization problems are viewed in a multiple decision-maker context (see, e.g., [17], [20]).

REFERENCES

[1] A. Agnetis, E. Grande, and A. Pacifici. Demand allocation with latency cost functions. Mathematical Programming, 132(1-2):277–294, 2012.

[2] Chawis B., Mikiharu A., and Takumi A. Facility location optimization model for emergency humanitarian logistics. International Journal of Disaster Risk Reduction, 24:485 – 498, 2017.

[3] B. Balcik and B. M. Beamon. Facility location in humanitarian relief. International Journal of Logistics Research and Applications, 11(2):101–121, 2008.

[4] J. Barceló and J. Casanovas. A heuristic lagrangean algorithm for the capacitated plant location problem. European Journal of Operational Research, 15(2):212–226, 1984.

[5] J. E. Beasley. Lagrangean heuristics for location problems. European Journal of Operational Research, 65(3):383–399, 1993.

[6] J. E. Beasley. Lagrangean heuristics for location problems. European Journal of Operational Research, 65:383–399, 1993.

[7] D. Bertsimas, V. F. Farias, and N. Trichakis. The price of fairness. Operations Research, 59:17–31, 02 2011.

[8] Z. Drezner and H. W. Hamacher. Facility location: applications and theory. Springer Science & Business Media, 2001.

[9] G. Falcone, G. Nicosia, and A. Pacifici. Minimizing part transfer costs in flexible manufacturing systems: A computational study on different lower bounds. In 2013 UKSim 15th International Conference on Computer Modelling and Simulation, pages 359–364, 2013.

[10] M. R. Garey and David S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, 1979.

[11] J. W. Gavett and Norman V. Plyter. The optimal assignment of facilities to locations by branch and bound. Oper. Res., 14(2):210–232, April 1966.

[12] K. Holmberg, M. Rönnqvist, and D. Yuan. An exact algorithm for the capacitated facility location problems with single sourcing. European Journal of Operational Research, 113(3):544–559, 1999.

[13] K. Holmberg, M. Rönqvist, and D. Yuan. An exact algorithm for the capacitated facility location problems with single sourcing. European Journal of Operational Research, 113:544–559, 1999.

[14] J. Klincewicz and H. Luss. A lagrangian relaxation heuristic for capacitated facility location with single-source constraints. Journal of the Operational Research Society, 37(5):495–500, 1986.

[15] M. Leitner and G. R. Raidl. Variable neighborhood search for a prize collecting capacity constrained connected facility location problem. In Proceedings of the 2008 International Symposium on Applications and the Internet, pages 233–236. IEEE Computer Society, 2008.

[16] M. Leitner and G. R. Raidl. Branch-and-cut-and-price for capacitated connected facility location. Journal of Mathematical Modelling and Algorithms, 10(3):245–267, Sep 2011.

[17] C. Marini, G. Nicosia, A. Pacifici, and U. Pferschy. Strategies in competing subset selection. Annals of Operations Research, 207(1):181–200, 2013.

[18] M. Naldi, G. Nicosia, A. Pacifici, and U. Pferschy. Profit-fairness trade-off in project selection. Socio-Economic Planning Sciences, https://doi.org/10.1016/j.seps.2018.10.007, 2018.

[19] G. Nicosia and A. Pacifici. Optimal allocation plan for distribution centres of a frozen food company. International Journal of Applied Management Science, 4(3):224–238, 2012.

[20] G. Nicosia, A. Pacifici, and U. Pferschy. Competitive subset selection with two agents. Discrete Applied Mathematics, 159(16):1865–1877, 2011.

[21] S. H. Owen and M. S Daskin. Strategic facility location: A review. European journal of operational research, 111(3):423–447, 1998.

[22] H. Pirkul. Efficient algorithms for the capacitated concentrator location problem. Computers & Operations Research, 14(3):197–208, 1987.

[23] R. Sridharan. A lagrangian heuristic for the capacitated plant location problem with single source constraints. European journal of operational research, 66(3):305–312, 1993.

[24] Sae yeon R., Hyun mi J., and Chul hwan H. Warehouse location decision factors in humanitarian relief logistics. The Asian Journal of Shipping and Logistics, 29(1):103 – 120, 2013.

TABLE I: HARD INSTANCE CLASSES WITH N = 30 AND M = 200.

#	f_i	k_i	d_j	p_{ij}
1	50	20	[5, 10]	20
2	50	20	[7, 8]	20
3	50	50	[5, 10]	20
4	50	50	[7, 8]	20
5	100	20	[5, 10]	20
6	100	20	[7, 8]	20
7	100	50	[5, 10]	20
8	100	50	[7, 8]	20
9	50	20	[5, 10]	[20, 50]
10	50	20	[7, 8]	[20, 50]
11	50	50	[5, 10]	[20, 50]
12	50	50	[7, 8]	[20, 50]
13	100	20	[5, 10]	[20, 50]
14	100	20	[7, 8]	[20, 50]
15	100	50	[5, 10]	[20, 50]
16	100	50	[7, 8]	[20, 50]
17	[6000, 8000]	600	[10, 50]	[100, 500]
18	[6000, 8000]	600	[10, 50]	[200, 400]
19	[6000, 8000]	600	[20, 40]	[100, 500]
20	[6000, 8000]	600	[20, 40]	[200, 400]
21	[6000, 8000]	[590, 610]	[10, 50]	[100, 500]
22	[6000, 8000]	[590, 610]	[10, 50]	[200, 400]
23	[6000, 8000]	[590, 610]	[20, 40]	[100, 500]
24	[6000, 8000]	[590, 610]	[20, 40]	[200, 400]
25	[6000, 8000]	[500, 700]	[10, 50]	[100, 500]
26	[6000, 8000]	[500, 700]	[10, 50]	[200, 400]
27	[6000, 8000]	[500, 700]	[20, 40]	[100, 500]
28	[6000, 8000]	[500, 700]	[20, 40]	[200, 400]
29	[6000, 10000]	600	[10, 50]	[100, 500]
30	[6000, 10000]	600	[10, 50]	[200, 400]
31	[6000, 10000]	600	[20, 40]	[100, 500]
32	[6000, 10000]	600	[20, 40]	[200, 400]
33	[6000, 10000]	[590, 610]	[10, 50]	[100, 500]
34	[6000, 10000]	[590, 610]	[10, 50]	[200, 400]
35	[6000, 10000]	[590, 610]	[20, 40]	[100, 500]
36	[6000, 10000]	[590, 610]	[20, 40]	[200, 400]
37	[6000, 10000]	[500, 700]	[10, 50]	[100, 500]
38	[6000, 10000]	[500, 700]	[10, 50]	[200, 400]
39	[6000, 10000]	[500, 700]	[20, 40]	[100, 500]
40	[6000, 10000]	[500, 700]	[20, 40]	[200, 400]
41	7000	600	[10, 50]	[100, 500]
42	7000	600	[10, 50]	[200, 400]
43	7000	600	[20, 40]	[100, 500]
44	7000	600	[20, 40]	[200, 400]
45	7000	[590, 610]	[10, 50]	[100, 500]
46	7000	[590, 610]	[10, 50]	[200, 400]
47	7000	[590, 610]	[20, 40]	[100, 500]
48	7000	[590, 610]	[20, 40]	[200, 400]
49	7000	[500, 700]	[10, 50]	[100, 500]
50	7000	[500, 700]	[10, 50]	[200, 400]
51	7000	[500, 700]	[20, 40]	[100, 500]
52	7000	[500, 700]	[20, 40]	[200, 400]

TABLE II: RESULTS ON OBJECTIVE FUNCTION VALUE.

	ILP	Heur-Assign		Mix-Assign $\alpha = 75\%$	
Class	Opt	Obj	Gap%	Obj	Gap%
1	460	358	21.74%	358	21.74%
2	0	−10	10*	−10	10∗
3	2490	2396	3.79%	2400	3.63%
4	2400	2376	1.66%	2376	1.66%
5	0	−20	20*	−20	20∗
6	0	−60	60*	−60	60∗
7	992	t.l.	–	t.l.	–
8	900	872	5.15%	872	5.15%
9	3302.4	2850.8	13.67%	2850.8	13.67%
10	1499.2	1484.2	0.99%	1484.2	0.99%
11	8322	7742.8	6.96%	7852.4	5.64%
12	8202.5	7642.8	6.93%	7642.8	6.93%
13	1747	1420.6	26.90%	1420.6	26.9%
14	0	t.l.	–	t.l.	–
15	6790.8	6242.4	8.09%	6343.4	6.6%
16	6646	6119.6	8.56%	6119.6	8.56%
17	31052.5	28972.8	4.12%	29042.4	4.08%
18	15749	16151.4	4.49%	16146.2	4.56%
19	–	25650.8	–	25947.2	–
20	14162.33	12637.8	9.84%	12698.6	10.2%
21	27953	28018	11.39%	28288.6	9.16%
22	17160	16179.8	3.68%	16236.4	3.92%
23	29315.5	25436.2	12.05%	26364.6	8.38%
24	12838.5	11740.4	8.73%	12012	8.52%
25	32961	29958	6.54%	31111.6	2.06%
26	20314	19549.2	4.56%	19834.8	3%
27	–	27424.8	–	27605.6	–
28	16506.2	15105.6	8.37%	15501.8	6.09%
29	27130.75	26707.2	3.58%	27101	2.61%
30	16331.8	15747	3.60%	15718.8	3.76%
31	28216	25921.4	5.22%	25593.2	8.65%
32	11200	11364.2	8.53%	11156	9.78%
33	27928.4	26873	3.76%	26981.2	3.39%
34	16194.2	15553.8	4.12%	15553.8	4.12%
35	26346	23543.6	8.64%	24402.8	3.5%
36	11507.33	10520.2	7.67%	10516.2	7.72%
37	30301	28646.8	3.93%	28871	3.04%
38	17159.6	16474.6	3.95%	16570.8	3.51%
39	28709	25811.6	6.49%	26694.6	4.93%
40	15211	13415	7.77%	13333	8.53%
41	–	24288.8	–	24960.6	–
42	15786	14107.6	7.15%	14107.6	7.15%
43	–	20944.4	–	21640.4	–
44	–	8708	–	8460.8	–
45	–	24770.8	–	24902	–
46	13512.5	11897.8	11.86%	11897.8	11.86%
47	–	21355.2	–	21920.6	–
48	–	8196.8	–	8109.8	–
49	31551	28030.8	6%	28694.8	2.89%
50	17609.33	16853.2	5.06%	16963.2	4.14%
51	28695.5	25389	15.21%	24765.2	16.16%
52	12043	12681.4	7.12%	13015	7.77%

TABLE III: CPU TIME COMPARISON

Class	ILP	Heur-Assign	Mix-Assign
1	8464.39	2.76	15.68
2	0.47	1.73	22.51
3	3.34	0.56	2.25
4	84.15	0.57	2.13
5	1.34	2.19	22.93
6	0.5	1.76	22.76
7	1.38	t.l.	t.l.
8	71.42	0.63	2.31
9	794.34	0.87	3.44
10	0.36	0.53	2.08
11	86.89	0.68	3.58
12	331.59	0.49	2.15
13	19.99	1.51	6.38
14	0.4	t.l.	t.l.
15	82.03	0.61	3.97
16	118.62	0.49	2.17
17	1177.67	2.2	93.37
18	521.21	2.09	79.61
19	–	1.91	40.78
20	1239.88	2.02	141.36
21	12562.59	2.11	98.04
22	3396.87	1.96	98.36
23	37.35	1.87	48.79
24	61.65	2.17	95.5
25	1499.59	2.24	138.06
26	497.14	2.07	133.42
27	–	1.76	25.36
28	772.67	2.09	75.32
29	765.78	2.3	143.06
30	86.76	2.2	63.4
31	134.17	2.23	69.64
32	38.59	2.16	84.85
33	965.03	2.19	113.21
34	83.29	2.08	87.48
35	660	2.14	83.28
36	88.55	2.22	72.67
37	3092.17	2.25	113.52
38	59.62	2.1	192.11
39	68.5	2.15	73.23
40	393.12	2.26	97.98
41	–	1.92	69.81
42	996.65	1.91	57.09
43	–	1.68	27.28
44	–	1.94	49.6
45	–	1.83	44.88
46	2237.17	1.86	181.11
47	–	1.75	29.5
48	–	1.86	56318
49	1641.48	2.2	102.16
50	571.66	2.03	95.5
51	811.62	2.01	30.6
52	39.41	2.21	109.46