AN APPROACH TOWARDS SOLVING LARGE-SCALE ALLOCATION PROBLEMS (MODELING AND SOLUTION ALGORITHM)

S. M. KHALESIZADEH, Ph.D.
Faculty of Industrial Engineering
Sharif University of Technology
Tehran, I. R. of Iran
email: khalessi@shrif.edu

A. BASIRIAN JAHROMI, M. S.
Faculty of Industrial Engineering
Sharif University of Technology
Tehran, I. R. of Iran
email: basirian@mehr.sharif.edu

Abstract - Large-Scale Allocation Problem (LSAP) is considered to be an important problem in Operations Research. In this paper, we have tried to model a sample LSAP and show that famous linear OR approaches in use cannot efficiently help us challenge, develop and handle such cases. We begin with efforts in this area to conclude that timely, efficient and less costly solutions could be only developed under simplified models. Then, a new approach will be introduced. We have tried to introduce an algorithm and applicative method to solve a sample LSAP model and to show that famous OR approaches, in use, cannot efficiently help us challenge, develop and handle such cases. We begin with model simplification, and conclude with a sample numerical solution of university entrance exam so that the efficiency of the method can be further explained.

Keywords – Large-Scale Allocation Problem (LSAP), Modeling Algorithm, Solution Algorithm, Linear OR Approaches, Operations Research.

INTRODUCTION

LSAP is a distinct problem branched from Allocation Problem also known as Assignment Problem. It is also a case of Linear Programming and Transportation Problem. However, while becoming large in scale, classical methods will prove insufficient in undertaking the model and solution.

LSAP is so applicative in various fields of technology and engineering that a great deal of different samples of its similar problems have been developed in recent years. Some practical and/or potential samples of LSAP, which have recently been worked on, will be mentioned below:

SOCIO-ECONOMIC APPLICATIONS

Allocating various industries to different geographical locations together with allocating required labor force to the established plants and/or industrial centers can be really valuable

to managers and decision makers. In a sample case in China, setting up agricultural-relevant plants and factories based upon resources, i.e. land, water, cattle, etc., has resulted in developing educational centers associated with agriculture and dairy along with having to carry and replace numbers of cattle.

HIGH-TECH APPLICATIONS

Allocating ISPs in an appropriate way for users not to get fed up with WAN and/or LAN's data transmission rates; or allocation of millions of transistors within ICs are two magnificent examples of LSAP, which have resulted in Network Engineering and VLSI (Very Large-Scale Integration) technology, respectively.

MILITARY APPLICATIONS

Allocation of different branches of military forces, Land, Air, and Navy, would certainly yield crucial LSAPs. In a sample case in Australian Air Force, certain aircrafts with wide ranges of military air power attributes, fighters, bombers, etc. were allocated into Australian airbases across the country so that air defense capabilities would be maximized concerning each and every airfield and airbase from the point of vulnerability to an imaginary potential invasion.

In another domestic case in Iran during war with Iraq, a semi-professional force of volunteers called "Basij" had desired to allocate different proficiencies of combat force so that the lost and/or wounded forces would be replaced as soon as possible, i.e. with the minimum delay. According to their combat proficiencies, which would vary diversely; e.g. gunmen, artillery, RPG men, anti-aircraft men, etc. and their number, this problem could be handled only as an LSAP.

OTHER APPLICATIONS

Assigning multitude of applicants (about 1.5 million) to the universities and/or fields of study applied for in the Nationwide Higher Education Entrance Exam along with assignment of teachers to various schools and educational centers in a large metropolitan area like Tehran, are other cases; the latter we will further develop as our sample model of an LSAP. Therefore, we define required variables, parameters and constraints accommodating the most realistic situation possible.

Solving an LSAP using available methods and algorithms is too difficult and time-consuming, if not impossible! In this paper, we present an algorithmic method which would be able to solve various types of such problems in a timely, practical and applicative

manner without affecting the generality of the problem and model. It is also possible to generalize this method; because, it is not limited to the number of parameters, restrictions, objective functions and so on. All such problems utilize a unified, general and integrated method to arrive at their appropriate solutions. Explicitly, through developing this method, more system flexibility and generality will result in contingency and adaptability. Ultimately, its application for practical instances is ever-enhanced in a dynamic rather than static way.

MODELING AN LSAP: TEACHER'S ASSIGNMENT TO SCHOOLS

We begin with a basic definition of parameters:
An M*N matrix of 0-1 elements that will define each teacher's capability of teaching any relevant course; represented by $\mathbf{T}_i^p$, where

p = 1,2, ..., M; number of teachers

i = 1,2, …, N; number of courses

Definitely,

$$\mathbf{T}_i^p = \begin{cases} 1; & \textit{if teacher p can teach course i} \\ 0; & \textit{otherwise} \end{cases} \quad (1)$$

Set of all schools in Tehran,

k = 1, 2, …, S; number of schools (2)

Set of official weekly teaching according to the educational calendar,
h = 1, 2, …, δ, where h = 1 and h = δ re present the first and last weekly educational teaching hours, respectively.
Set of schools' educational weekly schedule; a three dimensional matrix of integer elements; represented by $\mathbf{E}_{k,h}^{i}$, where

$\mathbf{E}_{k,h}^{i}$ = number of teachers required to teach course i in school k during educational hour h

Definitely, $\mathbf{E}_{k,h}^{i} >= 0.$ (3)

An M*S matrix provided with elements to show distances between each teacher's residence and each school; represented by $\mathbf{D}_k^p$, where

$\mathbf{D}_k^p$ = distance between teacher p's residence and school k (4)

It is now obvious that we have three matrices described in **(1)**, **(3)**, and **(4)** as our model basic inputs. The model output matrix, which would be our eventual solution to the problem and to be decided later, is actually a four dimensional matrix of 0-1 elements represented by $\theta_{k,h}^{p,i}$, where

$$\theta_{k,h}^{p,i} = \begin{cases} 1; & \textit{if teacher p would teach course i in school k at hour h} \\ 0; & \textit{otherwise} \end{cases} \quad (5)$$

Clearly, this output solution would be a sparse matrix of M*N*S* ä 0-1 elements !

To keep on with completing the model, now we introduce constraints of the problem:

(I) $\sum_{p} \theta_{k,h}^{p,i} \geq \mathbf{E}_{k,h}^{i}$

to supply number of teachers needed for teaching course *i* in hour *h* in any given school *k*.

(II) $\sum_{k} \theta_{k,h}^{p,i} \leq 1$

so that every teacher is confined to teach at one school at the maximum in any given hour *h*.

(III) $\alpha_1 \leq \sum \theta_{k,h}^{p,i} \leq \alpha_2$

to satisfy the minimum and maximum hours a teacher should teach every week according to the educational rules; where α_1 and α_2 are lower and upper bounds, respectively

(IV)
$$\forall\ \beta_1 \leq h \leq \beta_2\ ;\ k_1 \neq k_2 :\ \theta_{kl,h}^{p,i} * \theta_{k2,h}^{p,i} = 0\ \text{s} \quad \text{OR} \quad D_{k2}^{kl} \leq \gamma$$
$$\forall\ \beta_2 + 1 \leq h \leq \beta_3\ ;\ k_1 \neq k_2 :\ \theta_{kl,h}^{p,i} * \theta_{k2,h}^{p,i} = 0\ \text{s} \quad \text{OR} \quad D_{k2}^{kl} \leq \gamma$$
$$\cdots$$
$$\forall\ \beta_n + 1 \leq h \leq \beta_{n+1}\ ;\ k_1 \neq k_2 :\ \theta_{kl,h}^{p,i} * \theta_{k2,h}^{p,i} = 0\ \text{s} \quad \text{OR} \quad D_{k2}^{kl} \leq \gamma$$

to guarantee that teachers cannot be assigned to more than one school in some defined time interval, let us say between β_1 and β_2, because of distances difficulties; unless the distance between those schools is not farther than a distinct value, let us say γ. Clearly, β_i's should contend the following:

$\beta_1 = 1$, $\beta_{n+1} = \delta$, and that time intervals, e.g. β_2-β_1, should be an appropriate portion of time, let us say one complete educational day.

(V) $\theta_{k,h}^{p,i} \quad \mathbf{T}_i^p$

so that every teacher teaches courses of which s/he is capable. Finally, the object function of minimizing total distances traveled by teachers during a whole week, would come as follows:

(Object Function$_1$) $\mathbf{Min\ Z_1} = \sum_{p}\sum_{k} [(\theta_{k,h}^{p,i} / \sum \theta_{k,h}^{p,i}) * \mathbf{D}_k^p]$

As can be seen, this problem even in its present form, is a sophisticated non-linear problem that is not to be solved using well-known algorithms and methods, let alone adding more realistic constraints as we are keen to. It is, however, possible to develop a linear objective function using slack variables; but it would just result in a yet much larger

scale for our sample model.

MORE ELABORATE / GENERALIZED MODEL

In our primary model, we have relinquished some parameters and constraints to make it more convenient and simple to handle. Yet, its thorough completion needs other aspects to be taken into account. Therefore, we try here to describe them implicitly:

An M*N*S input matrix of teachers' preference/priority whose elements could vary over a fixed range of values, e.g. 0 to 10, (or is it better to be a fuzzy value?) is defined as:

$\Delta_{p,k}{}^{i}$ = teacher p's preference to teach course i in school k (6)

An M*N matrix of teachers' skill, proficiency and/or background in teaching courses, whose elements could be assumed in a similar way to **(6)**:

$\Pi_{p}{}^{i}$ = teacher p's proficiency/skill in teaching course i (7)

Now we add some extra constraints capable of being considered in the primary model:

(VI) $\forall\ k\ ,\ \theta_{k,h}{}^{p1,i}\ \ \theta_{k,h}{}^{p2,t}\ :\ \Delta_{p1,k}{}^{i} * \Pi_{p1}{}^{t}\ \ \Delta_{p2,k}{}^{i} * \Pi_{p2}{}^{t}$

so that no teacher with less qualification would replace his/her more qualified colleague in the same school. This is to certify that **(6)** and **(7)** play the main role in teachers' assignment to schools.

Some other constrains might be applied to our model a couple of which are:

1. Teachers whose $\Pi_{p}{}^{t}$ is less than some pre-defined value should not be allowed to teach in schools forcing a minimum of proficiency/skill.
2. Some advantages, i.e. increase in $\Pi_{p}{}^{i}$, could be delivered to those teachers who were volunteers to teach in schools with lower skills of total $\Pi_{p}{}^{i}$, i.e. school's proficiency of educational staff.

Hereafter, we can introduce our generalized model secondary object function:

(Object Function$_2$) Max $Z_2 = \sum \sum (\Delta_{p,k}{}^{i} * \sum \theta_{k,h}{}^{p,i})$

where we would aim at maximizing the total utility/satisfaction that teachers may obtain by teaching their most favorite courses in their favorite schools.

ALGORITHM'S APPROACH

– CATEGORIZING FACTORS HAVING AN INFLUENCE ON SELECTING PARAMETERS AND RESTRICTIONS

Pro factors: A factor is called pro while – on the condition that all other influential

factors in the model being supposed as constant – its quantitative increase causes an applicant/consumer to become more qualified to be given a definite/distinct resource. As an example, in assigning workers to machines in a job shop or plant, factors such as background, seniority, skill, rapidity and considerateness for any worker on a machine could be picked as a pro factor which – if considered quantitatively – can be combined to produce an overall factor, i.e. performance. Hence, it is possible to make one decisive factor out of combination of any number of pro factors within the model.

Con factors: A factor is called con while – on the condition that all other influential factors in the model being supposed as constant – its quantitative increase causes an applicant/consumer to become less qualified to be given a definite/distinct resource. For instance, in allocating aircraft *A* to airbase *B* in aircraft's allocation problem model, having no maintenance facility for a special kind of aircraft in a given airbase (which is actually a restriction to the problem) might be picked as a highly weighed con factor making assignment practically impossible.

Now, we go through with recognizing pro and con factors in the model of teacher's assignment to schools in Tehran:

$\mathbf{T}_i^p$ = teacher p's ability to teach course i

$\mathbf{E}_{k,h}^{\ i}$ = teachers required to teach course i in school k during educational hour h

$\mathbf{\Delta}_{p,k}^{\ i}$ = teacher p's preference to teach course i in school k

$\Pi_p^{\ i}$ = teacher p's proficiency/skill in teaching course i

Pro factors: $\mathbf{T}_i^p$, $\mathbf{E}_{k,h}^{\ i}$, $_{p,k}^{\ i}$, $_{p}^{\ i}$

$\mathbf{D}_k^p$ = distance between teacher p's residence and school k

$\mathbf{N}_k^p$ = teacher p's unwillingness to teach in school k

Con factors: $\mathbf{D}_k^p$, $\mathbf{N}_k^p$

DEFINING OBJECT/PERFORMANCE FUNCTION AND ALLOCATION ALGORITHM

After recognizing pro and con factors, we must define an object/performance function for applicants/consumers to be assigned to the resources considering factors extracted previously. This function is defined so that for any applicant/consumer there will be a list of values, calculated in accordance with pro and con factors, which is to be the main basis for assignments and allocations:

$$F(Q) = \prod_{i=1}^{n} Pi(Q) / \prod_{j=1}^{m} Cj(Q),$$

where, $P_i(Q)$ and $C_j(Q)$ are model's pro and con factors, respectively. Definitely, pro factors are always in proportional and con factors in reciprocal relation with the object/performance function. Obviously in the mentioned problem of teachers' allocation, object function would become:

$$F_{k,h}^{p,i}(Q) = T_i^p * E_{k,h}^i * \Delta_{p,k}^i * \Pi_p^i / D_k^p * N_k^p$$

ALLOCATION ALGORITHM'S METHOD TO ASSIGN APPLICANTS/CONSUMERS TO RESOURCES

The algorithm will arrive at its solution in the following sequence:

Preparation Phase- We define an n-dimensional matrix (n is the number of indices introduced in object function; herein n=4) whose cells value is to be calculated according to pre-defined criteria.

We also define an input table based on applicants' object/performance function value. This establishes the measure whether to accept or reject an applicant.

very cell in the above matrix is then valued in association with object function calculation for each applicant. The factors should be quantified as an input matrix (explained in Preparation Phase).

Execution Phase- A minimum value of object function would be available for every resource to be assigned with applicants/consumers. To do this, we have the number of resources demanded (here, the number of teachers needed to teach each course in any given school) and number of applicants/consumers intended to be allocated together with their respective priorities. Let the calculated values be F_1 for school 1, F_2 for school 2... and F_k for school k. Definitely, those applicants whose F is greater than F_i are qualified for school *i*.

Finishing Phase- The two phases of Execution and (if necessary) Iteration are repeated until there would be no resource left unallocated. Experience shows that this method converges to a stable state after a finite number of repetitions and (if necessary – as explained in Iteration Phase) iterations. The problem is said to be solved whenever no extra repetition would come into a new allocation.

Iteration Phase- In case our model induces or forces different applicants /consumers to have different weights for their pro and/or con parameters in their object function calculation, this algorithm will perform sufficient number of iterations so that any applicant/consumer is appropriately allocated. Hence, the solution will be arrived at and stabilized in a finite number of executive passes; i.e. no excessive pass would result in a new solution.

As can be seen, the system's goal, which is definitely to optimize the total and overall allocation of resources and is usually in contradiction with restrictions induced both by the model and applicants/consumers, is supplied in the most convenient way. In fact, our solution is rather optimum than maximum while simultaneously incorporating the two opposite factor categories of pro and con. Since the value of most cells in the elementary defined matrix is zero, the algorithm can come to an optimum solution very fast. Also, for all non-zero cells, the calculated object function value does certainly have a maximum

assignable value incorporating every defined restriction, so that every allocation throughout the algorithm will simultaneously consider priorities and restrictions giving our method an empowered advantage over those utilized before.

IMPLEMENTATION

For implementing this algorithm, we can simply use some extra tables and/or matrices to store allocated resources at various execution levels, so that the calculation volume over the main matrix is greatly reduced. One way is to store a pointer vector to those cells whose value is non-zero, since the values in the main matrix are mostly zero (as explained before). Now, we go through with a numerical model to illustrate how the algorithm works.

To describe the algorithm, we consider the large scale allocation problem of university entrance exam in Iran as a typical example. In university entrance exam, which is held annually throughout the country, applicants are classified into different major categories according to their high school branches of graduation. Each category is given its own entrance exam test of different courses – mathematics, physics, chemistry, literature and foreign language – and applicants are then ranked according to their rough (non-weighed) scores. Each field of study in universities performs its relevant weights over those courses to obtain their weighed scores for the applicants intending to enter them. The same thing is true with other LSAPs such as job-shop allocation problem. While a worker may need to have higher skill and/or more experience to work on some machines, s/he might not need those factors working on some others. In such cases, where different criteria and weights are allowed (or even necessary), the algorithm will have to go through ITERATION PHASE. However, the sample numerical problem, which is to be solved here, simply considers the weights of criteria to be unified and identical, which will bypass ITERATION PHASE.

Preparation Phase- We will have numerous fields of study together with their respective capacities that are liable for applicants to choose from as their favorite. Here, we assume the number of fields and their capacities as shown in Table 1. This table is considered as one of the problem's Input matrices.

Table1: Available fields and capacities.

Field ID	1	2	3	4	5	6	7	8	9
Field Capacity	8	12	5	10	6	11	4	6	8

The applicants' answer sheets will be corrected and their rough scores normalized according to the criteria weights explained before. For our sample 99 applicants, you can see their scores in different courses as well as their normalized total scores and rankings in Table 2. For more convenience, it is also assumed that scores are all weighed according to the

same criteria, which means no Iteration Phase needs to be performed. The assumed weights are 4,3,1,2, and 1 for the five courses, respectively.

Table 2: Applicants rough scores, normalized scores, and rankings.

Applicant No.	Score in Math	Score in Physics	Score in Chemistry	Score in Foreign Language	Score in Literature	Normalized Total Score	Rank
S 01	34	53	61	72	89	589	45
S 02	56	80	42	63	45	677	25
S 03	91	74	83	77	62	885	05
S 04	71	45	23	19	22	502	67
S 97	45	29	34	65	75	506	65
S 98	21	31	12	34	39	296	96
S 99	87	90	79	82	97	958	01

Each applicant is also given a form in which s/he chooses his/her priorities according to his/her own favor. (Table3) Here, priorities comply with Field IDs explained in Table1. This table is also another Input matrix for the model.

Table 3: Applicants favorite priorities. (application forms for different universities/fields)

Applicant No.	Priority1	Priority2	Priority3	Priority4	Priority5	RANK
S 01	7	8	5	1	3	45
S 02	7	9	3	8	4	25
S 03	6	5	4	1	7	05
S 04	1	6	4	8	9	67
S 97	8	4	7	9	6	65
S 98	3	8	5	7	1	96
S 99	7	1	8	6	2	0

In this simplified numerical model, Normalized Total Scores; i.e. sum of multiplication of each criterion rough score into its respective weight, are considered Pro factors. However, numbers of each favorite priority is a Con factor. The higher the score and the lower the number of priority, an applicant is more eligible to be allocated; i.e. applicant in Rank 01 (Applicant S99) and Priority1 is firstly assigned. The assignments take place in accordance to field's capacities so that each applicant is allocated to his/her highest favorite priority unless its capacity is fully occupied (Rank25 is allocated to his/her Priority 2 because his/her Priority1, i.e. Field No. 7, is fully allocated). The grayed PRIORITY indicates the Field ID in which the applicant is admitted and allocated according to the method; while the grayed RANK indicates applicants NOT admitted at all; i.e. the algorithm does not assign them to any of the fields meaning they are practically rejected.

Table 4: Applicants' ranking and their algorithm-selected priorities.

Rank	Priority 1	Priority 2	Priority 3	Priority 4	Priority 5
01	7	1	8	6	2
02	4	9	1	2	5
03	5	7	2	3	9
04	1	4	3	7	8
05	6	5	4	1	7
06	7	5	1	2	3
07	9	1	2	6	4
08	2	4	7	1	5
09	4	3	5	2	7
10	9	6	1	8	5
11	9	5	2	3	4
12	4	2	1	6	8
13	1	7	6	8	5
14	6	4	9	8	2
15	8	3	1	9	5
16	2	6	7	5	8
17	7	3	4	5	1
18	1	8	3	6	4
19	5	6	8	4	3
20	1	6	3	4	8
21	7	5	6	8	3
22	8	5	3	4	9
23	9	1	8	7	6
24	1	8	7	9	3
25	7	9	3	8	4
26	1	4	7	8	6
27	8	2	6	1	4
28	2	8	3	1	7
29	1	6	7	8	4
30	1	4	3	8	7
31	7	3	6	8	5
32	7	1	5	3	4
33	4	1	5	8	2
34	5	3	2	6	7
35	4	2	1	8	6
36	9	6	2	5	7

Rank	Priority 1	Priority 2	Priority 3	Priority 4	Priority 5
37	8	5	2	1	4
38	5	4	7	8	9
39	8	1	2	5	7
40	6	5	1	9	2
41	1	3	6	9	7
42	5	7	3	1	2
43	8	4	1	3	5
44	3	8	7	9	6
45	7	8	5	1	3
46	1	6	8	9	3
47	6	3	7	1	2
48	1	5	8	2	3
49	3	6	7	5	9
50	2	7	3	8	4
51	6	3	8	1	9
52	9	8	6	7	2
53	2	3	5	1	4
54	3	8	4	1	9
55	3	2	8	5	9
56	4	6	8	3	2
57	8	4	3	5	6
58	5	7	3	4	9
59	7	8	4	2	1
60	6	1	9	5	4
61	5	7	3	9	8
62	8	2	9	1	3
63	4	6	9	3	7
64	6	2	5	7	4
65	8	4	7	9	6
66	7	6	9	3	5
67	1	6	4	8	9
68	1	3	2	6	5
69	8	7	6	1	3
70	7	9	1	8	3
71	8	9	7	3	1
72	4	1	2	3	6

Rank	Priority 1	Priority 2	Priority 3	Priority 4	Priority 5
73	6	2	7	4	3
74	6	7	5	8	1
75	3	9	5	8	1
76	5	3	4	6	1
77	7	5	4	9	6
78	4	9	8	7	1
79	8	4	5	6	3
80	7	2	4	3	6
81	1	9	6	2	3
82	8	9	5	1	6
83	1	3	7	8	5
84	9	2	3	1	5
85	9	4	3	1	5
86	5	2	1	7	8
87	6	1	8	4	2
88	2	7	3	6	9
89	5	2	8	4	7
90	9	2	6	7	3
91	3	2	4	7	9
92	2	1	7	4	3
93	6	1	9	2	4
94	1	7	6	9	4
95	4	5	9	6	3
96	3	8	5	7	1
97	4	6	8	7	5
98	4	7	8	6	9
99	5	1	2	7	3

The allocation is continued until either all fields are allocated with applicants or there is no applicant left; i.e. we come to the end of the applicants list. However, in practice, the former occurs almost always. The allocated applicants' rank to each field of study is shown in Table5.

Table 5: Applicants' assignment to the fields is finished.

Field ID	Field Capacity													
1	8	4	13	18	20	24	26	29	30	*				
2	12	8	16	28	48	50	53	55	62	68	72	73	80	*
3	5	31	41	44	45	49	*							
4	10	2	9	12	33	35	54	56	57	58	59	*		
5	6	3	19	32	34	38	42	*						
6	11	5	14	40	46	47	51	60	63	64	65	66	*	
7	4	1	6	17	21	*								
8	6	15	22	27	37	39	43	*						
9	8	7	10	11	23	25	36	52		61	*			

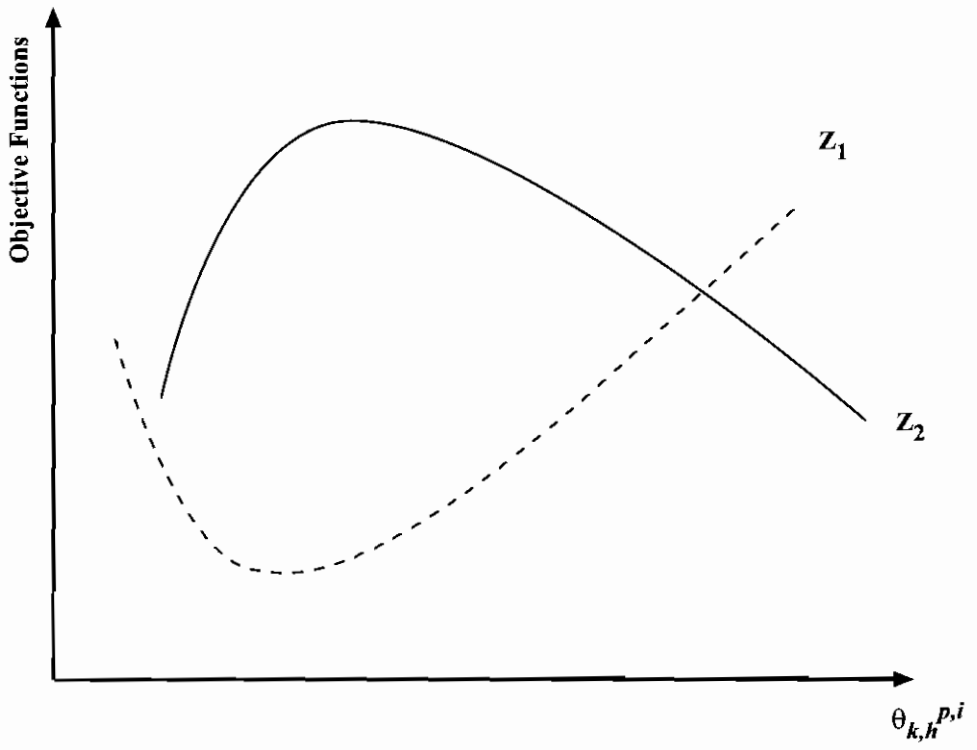
Step 4- The allocation algorithm is done; if for any applicant, the conditions are met i.e. the applicant is eligible from the point of his/her score and/or rank, and allocating him/her to that resource (here, the fields) has no constraint, then applicants substitution will be applied for his/her highest qualified priority as shown in Tables 4 and 5.

OTHER POTENTIAL APPLICATIONS: VIRTUAL AUCTION

This algorithm applies in many practical business and/or economic cases. A good example can be "Virtual Auction". In such a kind of auction, people do not need to be present at an auction's physical location. The auction manager will exhibit all goods due to go on sale. This will be the same as Field ID and Capacities (Table1). Similarly, people can apply for any of those represented goods by bidding some definite amount of money on any of their favorite items. This would be almost similar to Table 2 and/or Table3. The only difference is that each priority has its own amount of money bid on and will NOT be calculated. The algorithm, then, determines each item on the auction to go to the person who has bid the highest price on his/her highest priority.

CONCLUSIONS

As discussed in the last two sections, implementing the complete model may result in inconsistent objective functions of $\mathbf{Z}_1$ and $\mathbf{Z}_2$ as described before (Figure 1).



This makes our model more challenging; for, determining $(Z_1{*}\theta_1{*})$ or $(Z_2{*}\theta_2{*})$ solely is to be decided through hybrid heuristic methods and algorithms, let alone combining both objective functions into a trade-off solution, say $(Z_0{*}\theta_0{*})$. Hence, we first need to refine and redefine our model so that the generality of the problem will not be hurt. Then, we will need to develop an algorithm to make the solution time and process more economical. LSAPs are too difficult to arrive at solutions using ordinary methods. The presented algorithm, however, provides us with a timely and less complicated concept for LSAPs to be solved in a faster and more convenient way. By defining performance/objective functions one can easily enter new restrictions, parameters and variables in a previously solved problem and expect the new model to be solved so rapidly; i.e. generalization of any model can occur even if we have reciprocal and contrary goals to accomplish.

GENERAL REFERENCES

[1] Alian, C., et. al., "An Agent Based Architecture for Job-Shop Scheduling Problem Using the Spirit of Genetic Algorithm," Paris, 1999. *www.mit.jyu.fi/eurogen99/papers/vacher12.ps.*

[2] Armstrong, D. N., "Architectural Considerations for Network Processor Design," Univ. of Texas at Austin, 2002. *http://www.ece.utexas.edu/~bevans/courses/ee382c/projects/spring02/armstrong/LitSurveyReport.pdf,*

[3] Beck, J. C. and Refalo, Ph., "Combining Local Search and Linear Programming to Solve Earliness/Tardiness Scheduling Problem," *Proceedings CPAIOR,* 2002. *http://citeseer.nj.nec.com/cache/papers/cs/26330/beck02combining.pdf.*

[4] Goldengorin, B., et. al., "Equivalent Instances of the Simple Plant Location Problem," Univ. of Gorningen, 2000. *http://www.ub.rug.nl/eldoc/som/a/ 00A54/00A54.pdf.*

[5] Simpson, J. R. and Li, O., "Feasibility Study for Development of Northern China's Beef Industry and Grazing Lands," 1996. *http://uvalde.tamu.edu/jrm/nov96/ simpson.htm.*

[6] Wall, M. B., "A GA for Resource-Constrainced Scheduling," *MIT,* 1996. *http://lancet. mit.edu /~mbwall/phd/thesis/.*

[7] Walters, G. P., "A Defense System Development for Australian Air-Force," 1999. *http://handle.dtic.mil/100.2/ADA367990.*

[8] Yost, K. A., "Solution of Large–Scale Allocation Problems with Partially Observable Outcomes," 1998. *http://www.nps.navy.mil/orfacpag/resumePages/ papers/ washburnpa/LargeScale.pdf.*