A Preprocessing Procedure for Fixing the Binary variables in the Capacitated Facility Location Problem through Pairing and Surrogate Constraint Analysis

MARIA A. OSORIO[1], ABRAHAM SÁNCHEZ[2]
[1]Department of Chemical Engineering, [2]Department of Computing
Universidad Autónoma de Puebla
Ciudad Universitaria, San Manuel, Puebla. Pue.
MÉXICO
[1,2]{aosorio, asanchez}@cs.buap.mx http://www.cs.buap.mx/~{aosorio, asanchez}

Abstract: - The Osorio and Glover (2003) use of dual surrogate analysis is exploited to fix variables in capacitated facility location problems (CFLP). The surrogate constraint is obtained by weighting the original problem constraints by their associated dual values in the LP relaxation. A known solution is used to convert the objective function in a constraint that forces the solution to be less or equal to it. The surrogate constraint is paired with the objective function to obtain a combined constraint where negative variables are replaced by complemented variables and the resulting constraint used to fix binary variables in the model.

Keywords:- Capacitated Facility Location Problem, Surrogate Constraints, Duality, Constraint Pairing.

1 Introduction

We use dual surrogate constraint analysis to find the best surrogate and pairing it with the objective function, in the simplest case of capacitated facility location problems (CFLP), where there are m sources (or facility locations) which produce a single commodity for n customers each with a demand for d_j units (j = 1, ..., n). If a particular source is operating (or facility is built), it has a fixed cost $f_i \geq 0$ and a production capacity $K_i > 0$ associated with it. There is also a positive cost c_{ij} for shipping a unit from source i to a customer j. The question is where to locate the sources so that capacities are not exceeded and demands are met, all at a minimal total cost. All data are assumed to be integral.

The logistics for distribution of products (or services) has been a subject of increasing importance over the years. It is a significant part of the strategic planning of both public and private enterprises. Decisions concerning the best configuration for the installation of facilities in order to attend demand requests are the subject of a wide class of problems, known as *location problems*. These location problems have received a considerable amount of attention from scientists who have identified various problem types and developed variety methodologies to solve these problems, subsequently being adopted to make decisions belonging to locations of facilities in many practical applications. There are more than fifty facility location problem types (Lee and Yang, 2009). The location problems can be described as models in which a number of facilities is to be located in the presence of customers, so as to meet some specified objectives. Obvious applications of the problem occur when facilities such as warehouses, plants, hospitals, or fire stations are to be located. Although these instances are quite different from each other, they share some common features.

Most location problems can be defined as follows: given space, distance, a number of customers, customers' demands and mission. The distance is defined between any two points in that area. The number of customers is located in the area under consideration and who have a certain demand for a product (or service). The mission is to locate one or more facilities in that area that will satisfy some or all of the customers' demands. Depending on the objectives, location problems can be grouped into two major classes. One class treats the minimization of the average or total distance between customers and facilities. The classic model that represents the problems of this class is the p-median problem. Optimally locating public and private facilities such as schools, parks and distribution centers are typical examples of this problem. The other class deals with the maximum distance between any customer and the facility designed to attend the associated demand.

They often used in applications related the location of emergency facilities. These problems are known as covering problems and the maximum service distance is covering distance. The p-median is a well-known facility location problem which addresses the supply of a single commodity from a set of potential facility sites to a set of customers with known demands for the commodity. The problem consists of finding the locations of the facilities and the flows of the commodity from facilities to customers such that transportation costs are minimized. The combinatorial nature of the problem made it NP hard and encouraged many heuristic methodologies to approach the solution (Maric *et al.*, 2008).

To model the simplest case of the capacitated facilitated location problem (CFLP), we let x_{ij} be the amount shipped from source i to customer j, and define y_i to be 1 if source i is used and 0 if it is not. The integer programming model is:

(1) Minimize $z = \sum_{i\in M}\sum_{j\in N} c_{ij}x_{ij} + \sum_{i\in M} f_i y_i$

Subject to

(2) $\sum_{i\in M} x_{ij} \geq d_i$ $j \in N$ $(N = 1,2,\ldots,n)$
(3) $\sum_{j\in N} x_{ij} \leq K_i y_i$ $i \in M$ $(M= 1,2,\ldots,m)$
(4) $x_{ij} \geq 0$ $i \in M,\ j \in N$
(5) $y_i \in \{0,1\}$, $i \in M$

where $N = \{1,2 ,\ldots, n\}$, $M = \{1,2 ,\ldots, m\}$, $d_j \geq 0$, for all $j \in N$, $K_i \geq 0$, for all $i \in M$, $c_{ij} \geq 0$, for all $i \in M$, $j \in N$. This model has $m + n$ constraints and $m + mn$ variables. The objective function (1) is the total shipping cost, *i.e.*, $\sum_i \sum_j c_{ij} x_{ij}$, plus the total fixed cost, *i.e.*, $\sum_i f_i y_i$; note that f_i contributes to this sum only when $y_i = 1$ or source i is used. Constraints (2) guarantee that each customer's demand is met. Inequality (3) endures that we do not ship from a source which is not operating (K_i is an upper bound on the amount that may be shipped from source i) and it also restricts production from exceeding capacity. Even though by definition x_{ij} is discrete, we may use the nonnegative conditions (4) because it can be shown that constraint (5) with (2) and (3) ensure that $x_{ij} \geq 0$ will mean that x_{ij} is integer in the optimal solution (see Salkin (1975)).

The development of exact algorithms for integer problems began several decades ago (Dantzig (1957), Balas (1965), Glover (1965)). There exists a main stream of algorithms that try to find upper or lower bounds for the objective value, to reduce the problem size and use information from its relaxations, and to employ search trees in branch and bound schemes. The algorithm presented in this paper belongs in this mainstream.

The Constraint Pairing ideas used here were developed by Hammer *et al*. (1975) and used later by Dembo and Hammer (1980), as a support of a Reduction Algorithm for Knapsack problems that uses Constraint Pairing in a Lagrangean Relaxation framework. Glover established the main principles of his Surrogate Constraint Duality Theory in the same year (Glover, 1975), stimulating a series of algorithmic developments that used surrogate constraint analysis as an alternative to relying on weaker relaxations provided by Lagrangean Relaxation Theory. Recently, Glover *et al* (1997) generated cuts from surrogate constraint analysis for zero-one and multiple choice programming. Those cuts proved to be effective and stronger than a variety of those previously introduced in the literature.

In a different application, Osorio and Gómez (2004) used Surrogate Analysis in Multidimensional Knapsack Problems to create surrogate constraints and Constraint Pairing to combine them with the objective function to generate new constraints used to fix variables and to generate logic cuts, using an initial feasible integer solution. These logic cuts were included in the model before solving the problem with branch and bound. They tested in the set of small problems and big instances in the OR-library, and compared the effect of the different sets of logic cuts added to the model. It could be seen that any set of logic cuts helps to solve difficult problems with a fewer number of nodes in the search tree. Our procedure augments a branch and cut framework, by a process of fixing variables and adding global cuts. The approach can be applied every time the branch and cut method gets a better integer solution or it can be used as a preprocessing algorithm, based on assuming a bound on an optimal objective value. Their computational experiments showed that the preprocessing approach created an enhanced version of the problem that could be solved in a fewer number of nodes.

This paper applies the experience obtained by Osorio et al (2002), (2003) using surrogate constraint analysis to fix binary variables in MKP, to Capacitated Location Problems. The topic is presented in the following way. Section 2 presents the results on surrogate constraint duality that we apply in this paper and section 3 develops the constraint pairing ideas applied to this specific problem, according to the methodology described. Section 4 presents and example to illustrate these ideas, section 5, the computational results, and section 6, the conclusion.

2 Surrogate Constraint Analysis

Duality theory in mathematical programming is not new. For years, it has customarily been based upon the use of a generalized Lagrangean function to define the dual. Elegant results have emerged linking optimality conditions for the dual to those for the primal. Out of these results have arisen solution strategies for the primal that exploit the properties of the primal dual interrelations. Some of these strategies have been remarkably successful, particularly for problems in which the duality gap – the amount by which optimal objective function values for the two problems differ̃ is nonexistent or small. A different type of solution strategy has been proposed for solving mathematical programs in which duality gaps are likely to be large.

In contrast to the Lagrangean strategy, which absorbs a set of constraints into the objective function, a different strategy proposed by Glover (1975) replaces the original constraints by a new one called a surrogate constraint. Since their introduction by Glover (1965), surrogate constraints have been proposed by a variety of authors for use in solving nonconvex problems, especially those of integer programming. Surrogate constraints that were 'strongest' for 0-1 integer programming under certain relaxed assumptions were suggested by Balas (1967) and Geoffrion (1969). The paper by Geoffrion also contained a computational study that demonstrated the practical usefulness of such proposals. Later, Dyer (1980) provided a major treatment of surrogate duality. Methods for generating strongest surrogate constraints according to other definitions, in particular segregating side conditions and introducing normalizations, were subsequently proposed by Glover (1968). However, a significant price was paid by the relaxations used in some of these early developments, whose effect was to replace the original nonconvex surrogate IP problem by a linear programming problem. The structure of this LP problem is sufficiently simple that the distinction between the surrogate constraint approach and the Lagrangean approach vanished.

The first proposal for surrogate constraints (Glover, 1965) used notions that were later used in surrogate duality theory. It defined a strongest surrogate the same way as in this theory and presented a theorem that led to a procedure for searching optimal surrogate multipliers that could obtain stronger surrogate constraints for a variety of problems. Later, Greenberg and Pierskalla (1970) provided the first major treatment of surrogate constraints in the context of general mathematical programming. These authors showed that the dual surrogate is quasiconcave, thus assuring that any local maximum for it is a global maximum, and noted that the surrogate approach has a smaller duality gap than the Lagrangean approach. They provided sufficient conditions for the nonoccurrence of surrogate duality gaps.

The work of Glover (1975) developed a surrogate duality theory that provides exact conditions under which surrogate duality gaps cannot occur. These conditions (both necessary and sufficient) are less confining that those governing the absence of Lagrangean duality gaps. Furthermore, they give a precise characterization of the difference between surrogate and Lagrangean relaxation, and give a framework for combining these relaxations. Useful relationships for combining these relaxations are also developed in Karwan and Rardin (1979).

The primal problem of mathematical programming can be written:

P: $\min_{x\in X} f(x)$, subject to $g(x) \le 0$.

Where f and each component $g_i(x)$ of the vector $g(x)$ are real-valued functions defined on X. No special characteristics of these functions or of X will be assumed unless otherwise specified.

A surrogate constraint for P is a linear combination of the component constraints of $g(x)\le 0$ that associates a multiplier u_i with each $g_i(x)$ to produce the inequality $ug(x) \le 0$, where $u=(u_1, \dots u_m)$. Clearly, this inequality is implied by $g(x) \le 0$ whenever $u \ge 0$. Correspondingly, we define the surrogate problem as:

SP(u): $\min_{x\in X} f(x)$, subject to $ug(x) \le 0$.

The optimal objective function value for SP(u) will be denoted by $s(u)$, or more precisely, as:

$s(u) = \inf_{x\in X(u)} f(x)$, where $X(u) = \{ x \in X : ug(x) \le 0\}$.

Since SP(u) is a relaxation of P (for u nonnegative), $s(u)$ cannot exceed the optimal objective function value for P and approaches this value more closely as $ug(x) \le 0$. Choices for the vector u that improve the proximity of SP(u) to P – *i.e.* , that provide the greatest values of $s(u)$ – yield strongest surrogate constraints in a natural sense, and motivate the definition of the surrogate dual:

SD: $\max_{u\ge 0} s(u)$.

The surrogate dual may be compared with the Lagrangean dual LD: $\max_{u \ge 0} L(u)$, where L(u) is the function given by $L(u) = \inf_{x\in X} \{f(x) + ug(x)\}$. It should be noted that $s(u)$ is defined relative to the set X(u), which is more restrictive than the set X relative to which the Lagrangean L(u) is defined. Also, modifying the definition of L(u) by replacing X with $X(u)$, while possibly increasing L(u), will nevertheless result in $L(u) \le s(u)$ because of the restriction $ug(x) \le 0$; that is, L(u) may be regarded as an 'underestimating' function for both the surrogate and primal problems.

Another immediate observation is that any optimal solution to the surrogate problem that is feasible for the primal is automatically optimal for the primal and no complementary slackness conditions are required, in contrast to the case for the Lagrangean. These notions have been embodied in the applications of surrogate constraints since they were first proposed. Taken together, they provide what may be called a 'first duality theorem' for surrogate mathematical programming.

The strong surrogate optimality conditions are necessary and sufficient for optimality, and they can be used in a variety of new inferences for surrogate constraints in mathematical programming. The methodology proposed here draws on these fundamental results.

3 Constraint Pairing for CFLP

The main ideas about constraint pairing in integer programming were exposed by Hammer *et al.* (1975). Based on the objective of getting bounds for most variables, the strategy is to pair constraints in the original problem to produce bounds for some variables.

Based on the results exposed about surrogate constraints, the dual surrogate constraint provides the most useful relaxation of the constraint set, and can be paired with the objective function. Multiplying the set of inequalities in (3) by –1, and using the generic name a_{ij} for all the coefficients in the sets (2), (3) and (4), and z_j as a generic matrix notation name for all the variables, the resulting surrogate is:

$$\sum_{i\in M+N} u_i \ (a_{ij} \ z_j \ ) \ \geq \ \sum_{i\in M+\mathrm{N}} u_i \ \ b_i, \ \ j \ \in M+MN.$$

Now, we define $s_i = \sum_{i\in M+N} u_i\,(a_{ij}\,x_j)$ and make the objective function less or equal to a known integer solution (UB). If we use the generic name gi for all the coefficients in the objective function, the paired constraint between the surrogate and the objective function will be,

$$\sum_{j\in M+MN} (g_j - s_j)\, z_j \leq \mathrm{UB} \ - \ \sum_{i\in M+N} u_i\, b_i\, .$$

Coefficients for this paired constraint can be positive, negative or zero. To be able to use this constraint to fix variables in both bounds, all coefficients must be positive or zero. We substitute $w_j = 1–z_j$ in the negative coefficients $(g_j\text{-}s_j)$ to get positive ones $(g_j\text{-}s_j)$'and add the equivalent value in the right hand side. The resultant constraint is,

$$\sum_{j\in M+MN,+} (g_j\text{-}s_j)z_j + \sum_{j\in M+MN,-} (g_j\text{-}s_j)'\, w_j \leq$$
$$\mathrm{UB}–\sum_{i\in M+N} u_i\, b_i + \sum_{j\in M+MN,-} (\tilde{g_j s_j})'.$$

An interesting property of the last constraint is that the positive coefficient values are the negative reduced costs for the variables in the LP solution and the negative values correspond to the negative dual values of their bounds in the LP solution. Besides, the value of $\sum_{i\in M+N} u_i\, b_i \sum_{j\in M+MN,-} (g_j\text{-}s_j)'$ is the optimal solution of the LP problem (LB), and the right hand side of this paired constraint becomes the difference between the upper bound that corresponds to the best know solution for the problem and the lower bound that corresponds to the LP solution (UB–LB). This resultant constraint is used to fix variables to zero or one,

$$\sum_{j\in M+MN,+} (g_j\text{-}s_j)z_j + \sum_{j\in M+MN,-} (g_j\text{-}s_j)'\, w_j \ \leq \mathrm{UB\text{-}LB}.$$

If coefficients $(g_j\text{-}s_j)$ of z_j are greater to the difference (UB-LB), those variables must be zero in the integer solution; if the coefficients $(g_j\text{-}s_j)$' of w_j are greater to the same difference, those variables must be one in the integer solution because its complement, w_j must be zero. Variables whose coefficients are smaller than the difference remain in the problem.

Because we depend on the gap UB-LB and LB can not be changed because it is the LP continuous relaxed solution of the problem, a lower UB given by the best integer solution known, can increase the number of integer variables fixed.

4 Example

To illustrate the procedure described, we will use a model (Figure 1) with three sources and two demand points:

The data for the example can be seen in Table 1.

The mixed integer model for this example is,

Minimize $250\,y_1 + 180\,y_2 + 170\,y_3 + 65\,x_{11} + 70\,x_{12} + 60\,x_{21} + 65\,x_{22} + 55\,x_{31} + 60\,x_{32}$

Subject To

$x_{11} + x_{12} \leq 13\,y_1$ or $13y_1\text{-}x_{11}\text{-}x_{12} \geq 0$
$x_{21} + x_{22} \leq 9\,y_2$ or $9y_2\text{-}x_{21}\text{-}x_{22} \geq 0$
$x_{31} + x_{32} \leq 8\,y_3$ or $8y_3\text{-}x_{31}\text{-}x_{32} \geq 0$
$x_{11} + x_{21} + x_{31} \geq 7$
$x_{12} + x_{22} + x_{32} \geq 8$
$x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32} \geq 0$
$y_1, y_2, y_3 \in \{0,1\}$

After solving the relaxed LP problem, the corresponding dual values are u_j = {15, 20, 25, 80, 85}. The LP solution of the relaxed problem (with $0 \le y_i \le 1$, i = 1, 2, 3) is 1210 (LB) and the variable values: y_2=.77778, y_3 = 1, x_{21} = 7 and x_{32} = 8, with all other variables equal to 0. This problem has a known integer solution of 1260 (UB). The objective inequality and the dual surrogate constraint, obtained multiplying each dual value by its respective constraint in the model, are:

$$250\, y_1 + 180\, y_2 + 170\, y_3 + 65\, x_{11} + 70\, x_{12} + 60\, x_{21} + 65\, x_{22} + 55\, x_{31} + 60\, x_{32} \le 1260$$
$$195\, y_1 + 180\, y_2 + 200\, y_3 + 65\, x_{11} + 70\, x_{12} + 60\, x_{21} + 65\, x_{22} + 55\, x_{31} + 60\, x_{32} \ge 1240$$

And the paired constraint is, $55y_1\text{-}30y_3 \le 1260\text{-}1240$. Replacing y_3 by w_3= $(1–y_3)$, we get,

$$55y_1+30w_3 \le 1260\text{-}1240+30=1260(\text{UB})–1210(\text{LB})= 50.$$

From $55\, y_1 + 30\, w_3 \le 50$, we can fix $y_1 = 0$. In this case, if $y_1 = 0$, then $x_{11} = x_{12} = 0$, according to the first constraint in the model.

5 Computational Results

We tested three sets in order to explore our methodology impact in sets obtained with different generators.

a. Data sets using Hooker's generator

We tested our approach with the generator presented by Hooker et al (1999) and used by Osorio et al (1999) to prove the logic cuts efficiency. The fixed costs were generated using a normal distribution with a mean of 200 and a standard deviation of 20; the variable costs, according to the function: 50+5*|*i*-*j*|, where *i* and *j* are the warehouse and demand point indexes, respectively. The right hand sides for the capacity warehouse constraints were obtained with a normal distribution, using the number of warehouses as a mean and the half of this value as a standard deviation. For the demand amounts, we used numbers sequentially generated in order to accommodate the ratio tested of total warehouse capacity to total demand.

Two sets were tested in order to explore the impact of different reasons methodology with total storage capacity to total demand. . For the first set in Table 2 we always started with a fixed number and for the second test in Table 3, we chose the number of warehouses –1 as the first demand value and incremented sequentially from it.

The first integer solution by CPLEX was used as the best integer solution.

Results for these problems are shown in Tables 2 and 3. It can be seen that the procedure proposed allows the binary variables to be fixed in a good percentage. The time in which the procedure was executed was virtually 0 seconds in all cases.

5.2 IFORC Data set

For the second part of the experiment, we used the instances generated by (IFORCF, 2006) as proposed by Cornuejols, and Thizy Sridharan (1991). Overall results are reported for 24 instances of problems with different sizes and ratio r of total capacity to total demand. The small instances tested had a range of 25 customers and 10 potential demand points to instances with 500 customers and 50 potential demand points, and a ratio of total capacity to total demand, equal to 1.5 to 3 as reported in the database used.

The coordinates of clients and storage sites were selected randomly in a square of 1000 X 1000 for these problems. The customer demands were generated with a uniform distribution in the range of [5,35], rounded to the next integer. The capacities of the warehouses were generated with a uniform distribution in the range [10,160], rounded to the next integer. Transportation costs were calculated using the Euclidean distance between the customer and the warehouse sites multiplied by 0.01. Operating costs of storage was zero and fixed costs were calculated according to the following equation: Fixed Cost$_j$ = U[0,90] + U[100,110]*SQRT(Warehouse_capacity), where U[a, b] is the uniform distribution in [a, b]. After generating these data, the capacities of the warehouses were adjusted so that the ratio of total capacity to total demand equaled the desired ratio of 1.5 and 3.0.

Results for the instances tested are reported in Table 4. The percentage of fixed binary variables with this procedure ranges from 0% to 60%. The difference in the number of binary variables fixed for the instances generated under the same conditions leads to closely assess the characteristics of the problem that may impact on the performance of the proposed methodology. In particular, instances T311 and T411 were closely examined. In Problem T311, the percentage of binary variables fixed is 0% and in problem T411, the percentage of fixed variables is 60%, although both instances were produced with the same generador. The CPU time needed to fix the vaalues of the binary variables was less than 2 seconds in both cases and less than 0.02 seconds in the remaining problems. The characteristics of the T311 and T411 problems can be seen in Table 5.

Although the main characteristic data of the two bodies appear to be similar, as the ratio of demand / capacity problem, the sum of coefficients in the function introduces a difference. The main elements on the right side of the inequality used to fix the variables are the solution LP (LB) and the best integer MIP-known solution (UB). In Problem T311, this difference is 1197.42 and 353.25 on the instance T411. If one gets the ratio of the coefficients in the objective function over the coefficients in the constraint and the difference UB-LB, the results are more significant. This ratio is 29.65 for the problem T311, and 115.8188 for problem T411. This analysis can be seen in Table 6.

An important observation is that the effectiveness of the proposed methodology depends primarily on the closeness of the entire solution and the LP optimal solution related to the sum of the coefficients in the objective function and the coefficients in the overall restriction.

Testing this set of data with different methodologies, it can be noticed that instances obtained from this generator can be resolved almost completely with the different types of cuts generated by CPLEX and almost no nodes in the tree branch and bound, noting that the obtained synthetically problems with generators yield instances that are usually very sensitive to very specific methodologies. The performance of this methodology is not affected by the problem size or other characteristics as the ratio of capacity or the generator used to obtain instances of trial, but is influenced by the quality of the entire solution known. This method only needs to solve a problem LP whenever the fixation procedure is applied and can be used as often as new integer solutions act as upper limits can be found. This allows the method to be used in combination with a tree branch and bound or other methodologies that can take advantage of the knowledge generated by the procedure. Knowing in advance the real value of any of the binary variables in the problem and to reduce its size, can be extremely useful in problems that can not be solved optimally with other methodologies.

For instances too difficult it can be used in combination with heuristics to generate solutions that can be used as part of this process set. With the setting of variables and reducing the size of the problem the heuristics can be used again to generate a better solution as many times as the procedure can fix more binary variables.

6 Conclusions

Our procedure solves a linear program to generate a surrogate constraint that can be paired with the objective function to fix a percentage of the binary variables in the problem. The results obtained seem to be a promising way to reduce the size of the searching branch and bound tree. The approach can be applied every time the branch and cut method gets a better integer solution or it can be used as a preprocessing algorithm (as we have done), based on assuming a bound on an optimal objective value. Our computational experiments show that the preprocessing approach creates an enhanced method that allows problems to be solved by constructing smaller search trees.

References:

[1] Balas, E. (1965). “An Additive Algorithm for Solving Linear Programs with Zero-One Variables”. *Operations Research* **13**, 517-546.

[2] Balas, E (1967). “Discrete Programming by the Filter Method”, *Operations Research* **19**, 915-957.

[3] Dantzig, G. B. (1957). “Discrete Variables Problems”, *Operations Research* **5**, 266-277.

[4] Dembo, R., and P. Hammer (1980). “A Reduction Algorithm for Knapsack Problems”, *Methods of O.R.* **36**, 49-60.

[5] Dyer, H.E. (1980). “Calculating surrogate constraints”, *Mathematical Programming* **12**, 225-278.

[6] Glover, F. (1965). “A Multiphase-Dual Algorithm for the Zero-One Integer Programming Problem”. *Operations Research* **13**, 879-919.

[7] Glover, F. (1968). “Surrogate Constraints”, *Operations Research* **16**, 741-749.

[8] Glover, F. (1971). "Flows in Arborescences", *Management Science*, **17**(9), 568-586.

[9] Glover, F. (1975). “Surrogate Constraint Duality in Mathematical Programming”. *Operations Research* **23**(3), 434-451.

[10] Glover, F., Sherali, H., and Y. Lee (1997). “Generating Cuts from Surrogate Constraint Analysis for Zero-One and Multiple Choice Programming”, *Computational Optimization and Applications* **8**, 151-172.

[11] Granot, F., and P. L. Hammer (1971). “On the use of boolean functions in 0-1 linear programming”, *Methods of Operations Research*, 154-184.

[12] Geoffrion, A. (1969). “An Improved Implicit Enumeration Approach for Integer Programming”. *Operations Research* **17**, 437-454.

[13] Greenberg, H. and W. Pierskalla (1970). “Surrogate Mathematical Programs”. *Operations Research* **18**, 924-939.

[14] Hammer, P., Padberg, M. and U. Peled (1975). “Constraint Pairing in Integer Programming”, *INFOR* **13**(1), 68-81.

[15] Hooker, J.N. (1994). “Logic-based methods for optimization”, in A. Borning, ed., Principles and Practice of Constraint Programming, *Lecture Notes*

in Computer Science **874**, 336-349.
[16] Hooker, J.N., and M.A. Osorio (1999). "Mixed Logical/Linear Programming". *Discrete Applied Mathematics* **96-97**, 395-442.
[17] Karwan, M. H. and R.L. Rardin (1979). "Some relationships between Lagrangean and surrogate duality in integer programming", *Mathematical Programming* **17**, 230-334.
[18] Jeroslow, R. E., and J. K. Lowe (1984). "Modeling with integer variables", *Mathematical Programming Studies* **22**, 167-184.
[19] Lee, W. and N. Yang (2009). "Location problems solving by spreadsheets", *WSEAS Transactions on Business and Economics*, **6**(8), 469-480.
[20] Maric, M., Tuba, M. and Kratica, J. (2008), "Parameter adjustment for genetic algorithm for two-level hierarchical covering location problem". *WSEAS Transactions on Computers.* **7**(6), 746-755.
[20] Osorio, M. A., and R. Mújica (1999). "Logic Cuts Generation in a Branch and Cut Framework for Location Problems". *Investigación Operativa*, **8**(1,2,3), 155-166.
[21] Osorio, M.A., Glover, Fred, and Hammer, Peter (2002). "Cutting and Surrogate Constraint Analysis for Improved MKP Solutions", *Annals of Operations Research* **117**, 71-93.
[22] Osorio, M.A., Glover, F. (2003). "Exploiting Surrogate Constraint Análisis for Fixing Variables in both bounds for Multidimensional Knapsack Problems". Preecedings of the Fourth Mexican International Conference on Computer Science. Edited by Edgar Chávez, Jesús Favela, Marcelo Mejía and Alberto Oliart. IEEE Computer Society. IEEE, pp. 263-267.
[23] Osorio, M.A., and Gómez, E. (2004). "Logic Cuts Effect and Dual surrogate Analysis for Multidimensional Knapsack Problems", *WSEAS Transactions on Systems*, **3**(4), 1333-1339.
[24] Salkin, M. (1975). Integer Programming, Adisson-Wesley Publishing Company, New York.

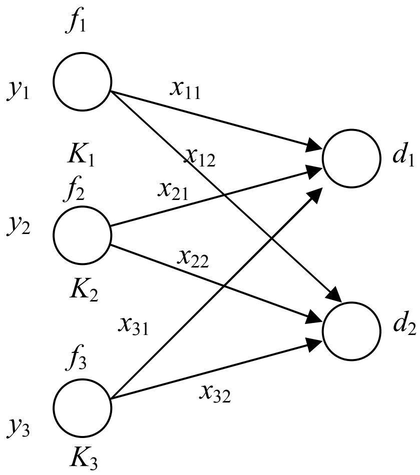

Fig. 1. Location model with three sources and two demand points.

Table 1. Data for the example

Fixed Costs for source i			Shipping costs by unit from source i to demand point j						Maximum on the amount from i			Demand units in j	
f_1	f_2	f_3	C_{11}	c_{12}	c_{21}	c_{22}	c_{31}	c_{32}	K_1	K_2	K_3	d_1	d_2
250	180	170	65	70	60	65	55	60	13	9	8	7	8

Table 2. Results for random instances with fixed initial demand

Warehouses	Demand Points	Number of Variables	Binary Variables	Binary Fixed to 0	Binary Fixed to 1	Ratio Capacity
20	5	120	20	13.9	0	8.8889
20	10	220	20	6.3	0	3.4783
20	15	320	20	3.1	0	1.9048
30	5	180	30	24.4	0	20
30	10	330	30	0	0	7.8261

Table 3. Results for random instances with initial demand equal to warehouses - 1

Warehouses	Demand Points	Number of Variables	Binary Variables	Binary Fixed to 0	Binary Fixed to 1	Ratio Capacity
20	10	220	20	7.4	0	2.5974
30	20	630	30	3.6	12.4	1.2594
40	30	1240	40	4.4	33	1.0774
50	40	2050	50	1.6	46.6	1.0001

Table 4. Results for the set of IFORCF instances

Name	Ratio	Warehouses	Demand Points	Number of Variables	Binary Variables	Solution		Fixed	
						LP	MIP	Variables	%
T111	1.5	10	25	260	10	33,993.01	34,684.31	2	20%
T211	1.5	10	25	260	10	35,341.58	35,999.81	2	20%
T411	1.5	10	25	260	10	52,338.89	52,692.14	6	60%
T511	1.5	10	25	260	10	22,583.63	23,443.57	3	30%
T512	3	10	25	260	10	16,187.24	18,582.39	1	10%
T121	1.5	25	50	1275	25	48,481.03	49,625.8	7	28%
T221	1.5	25	50	1275	25	57,553.97	58,100.14	4	16%
T321	1.5	25	50	1275	25	67,186.85	68,825.59	6	24%
T421	1.5	25	50	1275	25	55,446.97	56,313.57	9	36%
T521	1.5	25	50	1275	25	40,945.51	42,031.83	1	4%
T131	1.5	225	100	2525	25	75,558.07	78,368.85	3	12%
T231	1.5	25	100	2525	25	89,277.96	92,451.39	1	4%
T431	1.5	25	100	2525	25	69,930.24	72,756.81	1	4%
T531	1.5	25	100	2525	25	75,273.69	78,269.36	2	8%
T161	1.5	25	500	12525	25	38,3319.9	389,604.9	10	40%
T261	1.5	25	500	12525	25	31,7799.8	324,594.8	7	28%
T461	1.5	25	500	12525	25	33,9378.1	346,452.6	4	16%
T561	1.5	25	500	12525	25	29,4165.7	300,336.7	3	12%
T162	3	25	500	12525	25	30,3024	316,388.7	1	4%
T171	1.5	50	500	25050	50	252,088.4	260,612.2	1	2%
T271	1.5	50	500	25050	50	238,401.4	246,806.2	1	2%
T371	1.5	50	500	25050	50	263,744.1	273,241.9	1	2%
T481	1.5	50	500	25050	50	248,676.5	257,131.7	1	2%
T581	1.5	50	500	25050	50	237,806.3	247,536.1	3	6%

Table 5. Data Characteristics for instantes T311 and T411

Warehouses	T311 Capacity	Fixed Costs	Variable Costs			T411 Capacity	Fixed Costs	Variable Costs		
			Sum	Mean	Std.Desv.			Sum	Mean	Std.Desv.
1	26	546	2515	100.6	56.81	125	1217	4130	165.2	106.7
2	30	617	2436	97.45	54.44	47	791	2988	119.5	89.5
3	150	1299	2471	98.83	62.58	19	548	3096	123.8	84.22
4	88	1087	2274	90.97	50	114	1142	3590	143.6	94.8
5	106	1110	3503	140.1	71.97	90	1150	2984	119.4	71.62
6	77	971	2671	106.8	64.72	76	1028	2133	85.3	48.16
7	74	901	2904	116.2	65.15	76	943	2525	101	63.38
8	105	1110	2952	118.1	61.78	118	1255	3778	151.1	96.89
9	67	947	2372	94.89	51.55	21	529	3090	123.6	87.2
10	26	587	2983	119.3	65.29	134	1311	3505	140.2	92.95
Sum	749	9175	27081	1083	604.3	820	9914	31819	1273	835.4
Mean	74.9	917.5	2708	108.3	60.43	82	991.4	3182	127.3	83.54
Desv.	38.1456	242.9	357.2	14.28	6.637	39.9049	269.2	561.2	22.45	16.67

Table 6. Data Analysis for problems T311 and T411

Problem	T311	T411
Total Capacity	749	820
Total Demand	499	546
Total Capacity/ Total Demand	1.501002	1.501832
Total Sum RHS	-499	-546
Objective Coefficients Sum	36256	41733
Constraints Coefficients Sum	-749	-820
Objective Coefficients + Constraint Coefficients	35507	40913
LP Solution (LB)	30159.14	52338.89
Best Integer MIP (UB)	31356.56	52692.14
UB-LB	1197.42	353.25
(Objective Coefficients + Constraint Coefficients)/(UB-LB)	29.65292	115.8188