CHAPTER II

LITERATURE REVIEW

In this chapter, different methods for solving the capacitated p-median problem proposed in the literature are presented. First the works where exact methods are proposed are presented. Second, a set of works using heuristic approaches are presented.

In (Ceselli, Two exact algorithms for the capacitated p-median problem, 2003) two exact methods are presented for solving the CPMP: the first one generates a lower bound from a Lagrangean relaxation and an upper bound is found by first fixing the medians and getting a general assignment problem (GAP) and solving it by a branch and bound. The second method uses a Dantzig-Wolfe reformulation for the CPMP and by solving its linear relaxation as a set partitioning problem the dual bound is found, this reformulation is equivalent of the decomposition performed in the Lagrangean relaxation. The primal bound is then obtained by solving a GAP.

The use of column generation method is presented by Lorena and Senne (2004), this approach needs to transform the problem into a set covering problem which will be solved by column generation, and this is an example when the problem is reformulated into a different one in order to be solved. But if the columns are generated directly the convergence is slow so in order to speed up the algorithm a Lagrangean/surrogate relaxation is used, the assignment constraint is the one relaxed. The Lagrangean/surrogate introduces a second multiplier which will accelerate the overall process of finding lower bounds. This creates knapsack sub-problems. By solving those problems new columns are generated. (Lorena & Senne, 2004)

A cutting planes method using Fenchel cuts is proposed by (Boccia, Sforza, Sterle, & Vasilyev, 2008). With this approach the authors manage to reduce the integrality gap in hard instances and in some cases to find answers to previously unsolved hard instances. The formulation used in that work allows demand nodes to become medians and also allows a median to be assigned to another median. The authors also propose to reformulate the capacity constraint as knapsack constraints but this requires knowing the convex hull of the feasible solutions of the knapsack problem, in theory by solving a linear relaxation with the proposed formulation better lower bounds are obtained as opposed of doing a linear relaxation of the original formulation. As knowing the convex hull of feasible solutions is not easy there are two approaches: a Lagrangean relaxation or a Dantzig-Wolfe decomposition, the authors use the Fenchel cuts as they are dual to the decomposition and in theory will yield the same lower bound. (Boccia, Sforza, Sterle, & Vasilyev, 2008)

In the work of (Yaghini, Momeni, Sarmadi, & Ahadi, 2013)a local branching and a relaxation induced neighborhood search (RINS) is proposed. The local branching is an exact method often used to solve mixed-integer programs. Once a feasible solution is provided to start the algorithm; branching constraints are added. The RINS algorithm provides a solution using a continuous relaxation of an integer model based on a branch-and-cut tree. The relaxation is almost all the time not integral and the variables have to be corrected until integrality is met. If a better solution is found a new branch is created. When no feasible solutions are found the neighborhood is explored using two methods: first by either enlarging it, and if that fails a tabu constraint is introduced and a tabu search is performed. (Yaghini, Momeni, Sarmadi, & Ahadi, 2013)

Genetic algorithms are heuristic methods that mimic the process of evolution. A feasible solution is called a chromosome and each of the parts of this chromosome is called a gene. First an initial set of feasible answers for the problem is generated; this set of solutions is referred as a population. This population is then evaluated, usually using the cost function; the resulting values are how well each chromosome solves the problem and these results are called fitenss. This fitness will be used to assign a probability to each chromosome; those that have better results will have better probabilities. The chromosomes are randomly selected (the best ones have better chances of being selected) and then recombined to generate new solutions and a new population. Once the new chromosome is being conformed there is a mutation operation that randomly changes a part of it. Each iteration is called a generation. This method is adapted to the capacitated p-median problem in (Correa, Steiner, Freitas, & Carnieri, 2004). A chromosome is a set of size p which denotes what vertices are selected as medians; then an assignment algorithm is used to find the fitness. The crossover algorithm to generate new solutions does not allow medians to be repeated as that would be an unfeasible solution. The mutation changes a random median for another not present in the solution with a certain probability, usually small. The authors propose a hyper mutation heuristic, which happens when the initial population is created and after that only with a probability of 0.5% at each iteration. This heuristic takes a percentage of the solutions, for each of them a set of the unselected medians is created, and for each gene (median) all the unselected ones are tested, if any of them improves the current solution that median is then chosen. This greatly improves the fitness of those solutions that goes under this procedure. (Correa, Steiner, Freitas, & Carnieri, 2004)

The work of (Olivetti de França, Von Zuben, & Nunes de Castro, 2005) presents another bio-inspired algorithm for solving the capacitated p-median problem: the ant system that was proposed by Dorigo. This algorithm uses a pheromone trail as a reinforcement mechanism to allow smaller paths to be chosen, in the case of the CPMP the vertices of median that have the smallest values. The max min ant system improves the search mechanics by combining exploitation and exploration and also imposes bounds to the pheromone levels in order to avoid getting stuck in local optima, this version of ant system algorithm is the one used by the authors to solve the CPMP. Each of the ants chooses a series on nodes and opposed to the traveling salesman problem the pheromone is assigned to the nodes and not to the edges. After the medians are chosen using a general assignment problem the remaining nodes are allocated, this problem is solved by a constructive heuristic which creates lists of the distances to the medians from each node and the closest available one is selected. After this phase an improvement phase is performed, where the farthest nodes of each median are evaluated and a better median to assign them is searched; another step is to find better medians by comparing the best node to become a median and the worst median. Then the pheromone is updated and the most compact medians receive more pheromone and as such their chances of being selected by other ants are improved.

Similar to the previous approaches is the one proposed in (Yaghini, Lessan, & Gholami Mazinan, 2010). In this work a hybrid metaheuristic is used to solve the CPMP. A hybrid metaheuristic takes components from at least two algorithms and uses their characteristics to improve the exploration of the feasible region; this approach has become more popular in the recent years. In that work the main heuristic is a genetic algorithm and then it uses an ant colony algorithm characteristic: the pheromone trail. This trail is used in the assignment of each demand point to a median. When the algorithm starts the trail is the same for every demand point to any median; within each iteration the best median is selected and the pheromone trail gets stronger for shorter distances, those medians with higher pheromone values will have better chances of being chosen when assigning a demand point.

Maniezzo, Mingozzi and Baldacci (1998) present the use of a bionomic algorithm for a CPMP. The bionomic algorithm is an evolutionary algorithm that has as a base the genetic algorithm and improves upon it. In order to select the parents instead of selecting them by probabilities (where those with better objective function value have more chances of being selected) the parents are selected in a way that the most independent parents are chosen, this means that their medians are as different as possible, this is to assure variety in the population and to prevent getting stuck in local optima. Also a maturation step is taken after the new population is generated, in the case of the CPMP better medians are searched for each cluster, this step makes the overall process of optimization faster when compared with the genetic algorithm (Maniezzo, Mingozzi, & Baldacci, 1998)

Another hybrid heuristic is used in (Kaveh, Zadeh, & Sahraeian, 2010). They propose a k*-means* clustering algorithm to generate initial solutions and a fixed neighborhood search algorithm to perform a local search within each cluster previously formed. The k-means clustering is a learning algorithm, this method selects k nodes randomly and then assigns the closest customers to it to create the clusters, then new medians are calculated for each cluster and the process is repeated until the medians does not change anymore. For this problem k is the number of medians and therefore equals to p. This will be an initial solution of the FNS algorithm. This algorithm is modified to be suitable for the CPMP; some sites are eliminated as they would yield unfeasible solutions, also there is a tabu list added to avoid getting stuck in the same solutions. If the solution does not improve after certain number of iterations the algorithm is stopped.

Furthermore (Chaves, Correa, & Lorena, 2007) take the evolutionary clustering search which employs a clustering process that is executed at the same time with an evolutionary algorithm and generalizes it by making combinations with tabu search, variable neighborhood search or simulated annealing instead of the evolutionary algorithm. Each cluster is defined by a center, a radius and a search strategy that will be used in it. The cluster search has four components: a search metaheuristic that functions as a solution generator; an iterative clustering gathers similar solutions into groups and keeps a cluster center; an analyzer module which indicates what clusters are probable of yielding good solutions and a local searcher which is applied to the already identified clusters that are probable of yielding good solutions.

A different approach from the previously mentioned is the one described in (Shamsipoor, Sandidzadeh, & Yaghini, 2012) which uses a neural network to solve the CPMP. A neural network is a technique of machine learning which is used to minimize a cost function, usually in order to minimize errors. By using this approach for the CPMP the objective function is unconstrained and adds all the restrictions to it as penalties. This method reminds of the Lagrangean relaxation. In order to assign the customer nodes to the medians a dynamic heuristic is proposed.