

# A Genetic Algorithm for solving the Discrete Ordered Median Problem with Induced Order

Enrique Domínguez<sup>1</sup>, Alfredo Marín<sup>2</sup>

<sup>1</sup> Department of Computer Science,  
University of Málaga (Spain)  
enriqued@lcc.uma.es

<sup>2</sup> Department of Statistics and Operational Research,  
University of Murcia (Spain)  
amarin@um.es

**Abstract.** The Discrete Ordered Median Problem with Induced Ordered (DOMP+IO) is a multi-facility version of the classical discrete ordered median problem (DOMP), which has been widely studied. Several exact methods have been proposed to solve the DOMP, however these methods could only solve small-scale problems, which are far of real-life problems. In this work, a DOMP+IO with two kinds of facilities is considered and a heuristic method is proposed for its solving. The proposed procedure is based on a genetic algorithm and the preliminary results show the efficiency and capability to obtain good solutions for large-scale problems.

**Keywords:** Discrete ordered median problem, induced ordered, genetic algorithms.

## 1 Introduction

Despite the multitude of location problems and extensions that have been developed, it is possible, in some occasions to unify basic location problems that appear to be drastically different. A prominent recent development is the Discrete Ordered Median Problem (DOMP)[2], which is well known for its ability to generalize the  $p$ -median and  $p$ -center problems as well as define new problems based on the order of the demands in terms of their closest service distance or lowest cost. DOMP was introduced to provide a way to model many of the popular discrete location models, based upon ordering demand in terms of service by their respective closest facility. The main advantage of DOMP is the general structure, which is capable to unify the  $p$ -median problem and the  $p$ -center problem, as well as represent intermediate hybrid problems between median and center problems. This is achieved by introducing the order to the objective function and applying a set of penalties depending on this order, that is, by setting a vector  $\lambda$  of weight values on the demands according to their ranked distances to their closest facilities. Since the  $p$ -center problem measures the worst case service distance and the  $p$ -median problem measures the sum of the service distances, they can both be treated as a special case of DOMP with an appropriately defined  $\lambda$  vector. A great number of algorithms has been developed to solve the DOMP. Both exact methods[1] and heuristics[2] have been successfully applied to solve the problem. Moreover, specialized formulations have been also introduced to solve optimally medium sized instances of DOMP[3].

In this work, a multi-facility version of DOMP is addressed. Concretely, in this case two kinds of facilities are considered. Therefore, the goal of the proposed DOMP with induced order (DOMP+IO) is to locate all the facilities in such a way that a client should not be far from a secondary facility if she is far from (near to) a primary facility. For instance, hospitals can be considered as primary facilities and heliports as secondary facilities. Consequently, the generalization of DOMP is carried out by introducing an induced order on the secondary facilities. Preliminary results show that only small instances can be solved optimally in a reasonable computational time. In this work, a genetic algorithm is provided to attack larger instances of DOMP+IO.

## 2 Formulation for heuristics

The heuristic proposed to solve the large instances of the DOMP+IO is essentially based on genetic algorithms. In addition, evolution strategies are used to improve the performance of the proposed solver. Genetic algorithms use a vocabulary taken from natural genetics. In this sense, we talk about individuals or chromosomes in a population. A chromosome is divided into units, called genes. These genes contain information about one or several attributes of the individual, which represents a potential solution of the problem. The evolution in genetic algorithms can be implemented by two processes which mimic nature: natural selection and genetic change in the chromosomes or individuals, called reproduction. Natural selection consists of selecting those individuals that are better adapted to the environment, i.e. those who survive to achieve the best solution. Reproduction consists of genetic changes produced by genetic operators. Typically, there exists two basic genetic operators: crossover and mutation. The crossover transformation creates new individuals by combining parts from several (two or more) individuals. The mutation is an unary transformation, which creates new individuals by a small change in a single individual. After some generations the procedure converges and it is expected that the best individual represents a near-optimal solution.

The first task to create a genetic algorithm for solving a problem consists of specify a genetic representation of potential solutions of the problem.

### 2.1 Codification

An easy representation can be used according to the above proposed formulation for heuristic. Thus, an individual can be represented by two binary matrices (corresponding to allocations variables) and two binary vectors (corresponding to locations variables). In this sense, a classical binary codification can be used for the DOMP+IO. But two main disadvantages are uncovered:

- The classical binary operators almost never generate feasible solutions.
- The memory is inefficiently used because most of the positions contains a zero.

For these reasons, an integer codification is used for the DOMP+IO. Taken into account that the set of facilities is finite, the indexes are used to make up the representations. Then, an individual is composed by two integer vectors (corresponding to allocations variables) containing the indexes of the assigned facilities, and two integer vectors (corresponding to locations variables) containing the indexes of the elected facilities.

### 2.2 Initial population

A randomly method was considered to generate the initial population. At the first step, the facilities of each type were elected avoiding repetition inside the same type. That is, duplicate indexes are not permitted in the location part of the chromosome. However, the same index may occur in different blocks, i.e. a point can serves as facility of different types.

After of the facilities election, the allocation part of the individual is randomly filled according to the facilities selected at the previous step. That is, the permitted only values are the indexes of the elected facilities.

### 2.3 Evaluation function

Each individual of the population has associated a "fitness" value, which is determined by the objective function of the problem. Thus, the best solution of the problem is represented by the best adapted individual of the population, whom fitness is the best of the population.

## 2.4 Genetic operators

The genetic operators presented in this subsection are similar to the classical crossover and mutation operators[], but they are adapted to the actual codification in order to provide feasible solutions.

Two kind of crossover operators were considered, one by each type of facilities. Both operators are single point crossovers where only one block (first or second) of the chromosome is changed and the rest of the individual is conserved. The scheme for both crossovers is the same: a single breaking position is defined into the two parents individuals and the two children are generated by interchanging the right or left block-side of the breaking position. Afterwards, an adjustment procedure is carried out in order to avoid infeasible solutions. The possible duplicate indexes in the block are replaced by some prior values and the wrong indexes (corresponding to the old facilities) in the allocation part are replaced by the indexes of the new facilities. Note that the unique difference between the two provided operators is the block where the interchange is performed. Thus, a crossover operator is provided by each type of facilities.

In the same way, two kind of mutation operators are also considered, depending on the part of the individual to be changed. The mutation operators are based on the classical one for the binary codification: an index of the individual is selected and then it is changed by another one. If the selected index is in the allocation part of the individual, then this index is replaced by another one of the corresponding location part (allocation mutation). Otherwise (location mutation), the selected index of the location part is replaced by another non-used one (corresponding to the new facility). Afterward, a re-allocation process is carried out in order to avoid infeasible solutions.

## 2.5 Selection criterion

After the generation of the children, similarly to the natural selection in genetics, a selection criterion is applied, which depends on the evaluation function (fitness). Obviously, the number of individuals must be limited to avoid an overflowed population after several generations. In this sense, only the best generated children will form the new population. Furthermore, in order to preserve the best solutions between the generations, an elitism criterion is also considered. Note that both the number of individuals (population size) and the number of the survival individuals between generations (elitism) are common parameters of the genetic algorithm to be specified.

# 3 Experimental results

In order to test the proposed heuristic, several instances have been randomly generated with different combinations of the number of facilities and set of penalties. The preliminary results show the efficiency of the proposed genetic algorithm to provide good solutions for large-scale instances.

## References

- [1] Natasha Boland, Patricia Domínguez-Marín, Stefan Nickel, and Justo Puerto. Exact procedures for solving the discrete ordered median problem. *Computers and Operations Research*, 33(11):3270–3300, 2006.
- [2] Patricia Domínguez-Marín, Stefan Nickel, Pierre Hansen, and Nenad Mladenovic. Heuristic Procedures for Solving the Discrete Ordered Median Problem. *Annals of Operations Research*, 136(2005):145–173, 2005.
- [3] Alfredo Marín, Stefan Nickel, Justo Puerto, and Sebastian Velten. A flexible model and efficient solution strategies for discrete location problems. *Discrete Applied Mathematics*, 157(5):1128–1145, 2009.