Genetic algorithm for the capacitated plant location problem with single source constraints

Mitsuo Gen, Juno Choi and Yasuhiro Tsujimura

Department of Industrial and Information System Engineering
Ashikaga Institute of Technology
Ashikaga 326-8558, Japan
E-mail : {gen jochoi tsujimr}@genlab.ashitech.ac.jp

Abstract

Plant location models are applicable to problems in distribution systems, communication networks and other fields. In capacitated plant location problems, a number of plants with given capacities must be chosen from among a set of possible plant locations and then customers assigned to them. We describe a genetic algorithm for capacitated problems in which each customer is served by a single plant. In this paper, we present a completely new approach by using the genetic algorithm. The proposed approach has been tested on six test problems. Computational results are given.

KEYWORD: Location theory; Capacitated plant location; Genetic Algorithms.

1 Introduction

In this problem, we are given a set of potential location for plants with fixed costs and capacities. A commodity, for a set of customers with known demands, is to be supplied from these plants. The transportation cost for the commodity supplied from the plants to all customers is given. The problem is to find the subset of plants that will minimize the total fixed and transportation costs such that the demand of all the customers can be satisfied without violating the capacity constraints of the plants. The cost is the sum of the fixed costs of opening plants and the costs for assigning customers to specific plants which depend on, for example, the distance between them. The plant location problem can be classified into different categories depending on the restrictions assumed. In the uncapacitated plant location problem each plant is assumed to have no limits on its capacity. Here, each customer received all required service from one plant. When each plant has a limited capacity the problem is called the capacitated plant location problem.

The Capacitated Plant Location Problem with Single Source (CPLPSS) problem is a special case of the capacitated facility location problem in which each customer can only be supplied from one plant. This problem is generally more difficult to solve because all decision variables are integers as compared to the capacitated facility location problem where variables associated with customers are continuous.

In this paper, we propose a genetic algorithm based on the plant-based encoding and customer-based encoding for solving the CPLPSS. The procedure consists of two stages: (i) plant selection, and (ii) assignment. From this point, we design the plant-based encoding and customer-based encoding. The plant-based encoding is adapted as it is plant selection of stage 1, and the customer-based encoding is adapted as it is assigning customer of stage 2. We design the criterion for feasibility of the chromosome, and design genetic crossover and mutation operations based on criterion of feasibility. In order to improve the efficiency of the genetic algorithm, we utilize the reduced cost for optimality of a solution and hybridize to the genetic algorithm to avoid degeneration of the evolutionary process. Also, the mixed strategy that combined $(\mu + \lambda)$-selection and roulette wheel selection is used. The proposed algorithm can find the optimal solution for CPLPSS problem with short time in small-to-medium scale CPLPSS problem.

2 Problem Description

Neebe and Rao propose a branch and bound method for this problem for this problem. They formulate CPLPSS as a set partitioning problem and then consider the corresponding linear relaxation at each node of the enumeration tree to obtain bounds. They observe that the linear programing relaxation has a very high probability of terminating all integer and hence the branch and bound tree to solve CPLPSS is not very large [Neebe 1983].

Barcelo and Casanovas employ Lagrangian relaxation heuristic where they dualize the demand constraints [Barcelo 1984]. The first phase of the solution procedure defines a set of multipliers on the basis of an analysis of the dual linear program and selects a set of open plants. Once a set of plants is chosen, the problem is treated

as a special case of the generalised assignment problem and the assignment of customers to the open plants is carried out by an ad hoc modification to the Ross and Soland algorithm (1975).

Delmaire *et al* developed several heuristics, each based on one or more of the following approaches: evolutionary algorithms, greedy randomised adaptive search (GRASP), simulated annealing and Tabu search [Delmaire 1997].

CPLPSS can be formulated as follows.

Define

$$x_{ij} = \begin{cases} 1, \text{ if customer } i \text{ is assigned to plant } j, \\ 0, \text{ otherwise} \end{cases}$$

$$y_j = \begin{cases} 1, \text{ if plant } j \text{ is chosen}, \\ 0, \text{ otherwise} \end{cases}$$

and let

d_i = the demand of customer i,

s_j = the capacity of plant located at j,

c_{ij} = the cost of supplying all of the demand of customer i from a plant located at j,

f_j = the fixed cost associated with plant j,

Then,

$$\min \quad \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{j=1}^{n} f_j y_j \tag{1}$$

$$\text{s. t.} \quad \sum_{j=1}^{n} x_{ij} = 1 \quad \forall i \tag{2}$$

$$\sum_{i=1}^{m} d_i x_{ij} \leq s_j y_j \quad \forall j \tag{3}$$

$$\sum_{j=1}^{n} s_j y_j \geq \sum_{i=1}^{m} d_i \tag{4}$$

$$0 \leq x_{ij} \leq y_j \leq 1 \quad \forall i, j \tag{5}$$

$$x_{ij} \in \{0,1\} \qquad y_j \in \{0,1\} \quad \forall i, j. \tag{6}$$

Constraints (2) stipulate that each customer should be assigned to a single plant, Constraints (3) enforce the plant capacities, as well as ensure that a customer can be assigned only to a plant that is chosen. Constraints (4) stipulates that the total capacity of the chosen plants should be adequate to cover the total demand.

Constraint (4) is in fact redundant, since it is covered jointly by (2) and (3). However, it has been shown that adding such an aggregate capacity constraint leads to tighter lower bounds for CPLP. This is also the case for CPLPSS. Moreover, as will be shown later, adding this constraint facilitates the task of arriving at feasible solutions for CPLPSS.

An interesting feature of CPLPSS is its closeness to many other combinatorial problems. For example, when we take $f_j = 0$, for all j, we get the single source transportation problem. If $f_j = 0$ and if d_i is replaced by d_{ij} then we get the generalized assignment problem. If we take $c_{ij} = 0$, for all i, j, then we have the generalized bin packing problem (GBPP). When $d_i = 1$, for all i, and $s_j = k$, for all j, and $s_0 = n$, CPLPSS becomes a star-star concentrator location problem (SSCLP). Finally, when the single source restrictions are relaxed, we get the CPLP which is a fairly well known problem in the literature.

3 Genetic Algorithm Approach

3.1 Representation

The genetic representation is a kind of data structure which represents the candidate solution of the problem in coding space. Usually different data structures or genetic representations. Two representation schemes have been proposed for the CPLPSS problem.

- plant-based representation
- customer-based representation

position: plant ID	y_1	y_2	y_3	y_4	y_5
bit string	1	0	1	0	1

Figure 1: Plant-based representation

customer ID	1	2	3	4	5
open plant	1	3	5	1	3

Figure 2: Customer-based representation

Plant-based representation The plant-based representation is an obvious choice for the Phase 1 of plant selection since it represents the underlying 0-1 integer variables, *i.e.* for an n plants problem, an n-bit binary string can be used as the chromosome structure. A value of "1" for the ith plant implies that plant i is opened plant. An example is illustrated in Figure 1.

The fixed charge of an individual $f(x)$ is calculated simply by

$$f(x) = \sum_{j=1}^{5} f_j y_j \tag{7}$$

The initial population can be generated randomly. Note that individuals in the plant-based representation are not guaranteed to be feasible. It means that the total capacity of the opend plants not cover the total demand. We design the handling for feasibility of the initial population of the Phase 1 with the following criterion: The plant selection stage terminates when the total capacity of the open plants just exceeds the total demand.

The constructive procedure for the plant selection stage as follows:

Procedure: Plant selecting

step 1: Initialize plant-chromosomes randomly until the total capacity of the open plants just exceeds the total demand.

step 2: Check the feasibility constraint (4).

step 3: Sort the fixed cost for plants.

Customer-based representation The problem of maintaining feasibility (the total capacity of the open plants should be adequate to cover the total demand) may be resolved by using a customer-based representation. In this representation, the location of each gene corresponds to a customer and the encoded value of each gene is a opened plants that covers for total demand.

An example is illustrated in Figure 2, where plant 1 is covered customer, plant 2 is covered customer 1 and 3, and so on.

With this representation, feasibility can generally be maintained throughout the crossover and mutation procedure.

Procedure: Assigning customers

step1: Initialize customer-chromosomes.

step2: If $\sum_i d_i x_{ij} > s_j y_j$, $\forall j$ return to step 1. otherwise, continue.

step3: Calculate the transportation cost.

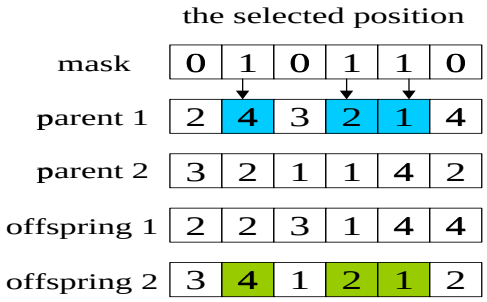

Figure 3: Illustration of uniform crossover

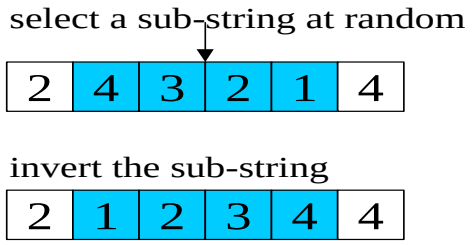

Figure 4: Illustration of inversion mutation

3.2 Genetic Operators-Crossover and Mutation

Crossover and mutation are two crucial factors in the biological evolutionary process. In this paper we adopt the uniform crossover operator which has been shown to be superior to traditional crossover strategies for combinatorial optimization problem. Uniform crossover firstly generates a random crossover mask and then exchanges relative genes between parents according to the mask. A crossover mask is simply a binary string with the same size of chromosome. The parity of each bit in the mask determines, for each corresponding bit in an offspring, which parent it will receive that bit form. The operation can be illustrated in Figure 3.

Mutation is a background operator, which produces spontaneous random changes in various chromosomes. A simple way to achieve mutation would be to alter one or more genes. Here the inversion mutation is used. The inversion mutation selects two positions within a chromosome at random and then inverts the substring between these two positions as illustrated in Figure 4.

3.3 Evaluation and Selection

As in nature, it is necessary to provide the possible driving mechanism for better individuals to survive. Evaluation is to associate each individual or chromosome with a fitness value that reflects how good it is based upon its achievement of the objectives. The higher the fitness value of an individual, the higher its chances of survival and reproduction and the larger its representation in the subsequent generation. Selection is to select individuals to be parents for the next generation according to their own fitness values. Therefore, the evaluation and selection play a very important role in the evolution process. The evaluation procedure can be operated as follows:

Procedure: Evaluation

step 1: Covert an individual into a tree.

step 2: Calculate the total cost and the total fixed cost of a tree according to the objective function.

step 3: Repeat the procedure on all individuals.

As to selection, we adopt the $(\mu + \lambda)$-selection and roulette wheel selection can enforce the best chromosomes into the next generation. This strategy selects μ best chromosomes from μ parents and λ offspring. If there are no μ different chromosomes available, then the vacant pool of population is filled up with roulette wheel selection [Gen 1999].

Table 1: Supplies, demand and costs for CPLPSS problem.

plants customers	1	2	3	4	5	6	7	8	9	10	demand
1	10	10	20	30	40	24	26	37	85	67	35
2	50	20	30	40	67	49	67	28	29	67	14
3	50	10	10	20	95	34	16	29	37	48	17
4	30	40	50	50	48	37	39	38	34	84	23
5	40	30	20	10	94	64	82	91	75	34	22
6	29	54	49	67	27	64	49	57	18	73	16
7	96	48	43	59	72	95	47	43	62	49	13
8	37	29	48	94	72	37	59	64	49	67	20
9	34	29	67	84	97	16	87	26	43	58	15
10	64	79	58	43	67	19	54	97	64	28	29
supply	50	50	50	50	50	50	50	50	50	50	
Fixed_charge	200	400	300	250	450	297	548	453	367	496	

Table 2: Average results by the proposed genetic algorithm.

problem size	parameter			optimal	CPU	Fixed	opened plant	chromosome
plants×customers	pop_plant	pop_cust	max_gen	solution	time (secs)	charge		
4×5	20	30	200	860	0.15	750	1,3,4	33414
5×6	20	30	200	889	0.3	750	1,3,4	334141
5×8	30	40	300	1350	0.65	1150	1,2,3,4	22314133
6×8	30	40	300	1357	0.65	1147	1,2,4,6	62414122
6×10	30	40	300	1678	2.15	1447	1,2,3,4,6	2241413663
10×10	40	50	500	1738	5.85	1514	1,2,4,6,9	1249492266
10×13	40	50	500	2248	9.67	1864	1,3,4,5,6,9	4935193966531

4 Numerical Experiment

The proposed genetic algorithm was implemented in C language and run on the 400MHZ IBM NT.

To confirm the effectiveness of the genetic algorithm for solving the CPLPSS problem, the eight numerical examples were used in the computational studies. These problems were generated as follows. The demands were generated from a uniform distribution in the range between 10 and 50. The capacities of the plants were generated 50. The fixed cost for plant j were generated from a uniform distribution in the range between 200 and 600. Table 1 shown the data of the example with 10 plants and 10 customers. For this example, the parameters for the algorithm are set as follows: crossover probability $P_C = 0.4$; mutation probability $p_m = 0.2$; and run time by 10 times. Table 2 shows simulation result from 10 times running by the proposed genetic algorithm.

5 Conclusion

The solution scheme presented genetic algorithm for solving the capacitated plant location problem with single source cosntraints proved powerful and capable of solving small-to-medium problem instances effectively in relatively short computation times.

We designed the criterion for feasibility of the chromosome. Based on this criterion the genetic crossover and mutation operations were designed and they can always generate the can always generate the feasible chromosomes. Whichever viewpoint is adopted, it seems that the combination could prove useful for many combinatorial optimization problems.

Acknowledgement

This research was supported by the International Scientific Research Program, the Grant-in-Aid for Scientific Research (No.10044173: 1998.4-2001.3) by the Ministry of Education, Science and Culture, the Japanese Government.

References

Hindi, K. S. and Pienkosz, K. (1999). Efficient solution of large scale, single-source, capacitated plant location problems, *Journal of the Operational Research Society*, 50, 268-274.

Sridharan. R. (1995). The capacitated plant location problem, *European Journal of Operational Research*, 87, 203-213.

Ronnqvist, M., Tragantalerngsak, S. and Holt, J. (1999). A repeated matching heuristic for the single-source capacitated facility location problem, *European Journal of Operational Research*, 116, 51-68.

Barcelo, J. and Casanovas, J. (1984). A heuristic Lagrangian algorithm for the capacitatd plant location probelm, *European Journal of Operational Research*, 15, 212-226.

Gen, M. and Cheng, R., *Genetic Algorithm and Engineering Design*, John Wiley and Sons, New York, 1997.

Gen, M. and Cheng, R., *Genetic Algorithm and Engineering Optimization*, John Wiley and Sons, New York, 1999.

Neebe, A. and Rao, M. (1983). An algorithm for the fixed-charge assigning users to sources problem, *J Opl Res Soc*, 34, 1107-1113.

Delmaire, D., Diaz, JA., Fernandez, E. and Ortega, M. (1997). Comparing new heuristics for the pure integer capacitated plant location problem, Research report DR97/10, Dpt E10, Univ Politecnica de Catalunya, Spain.