"Vasile Alecsandri" University of Bacău
Faculty of Sciences
Scientific Studies and Research
Series Mathematics and Informatics
Vol. 19 (2009), No. 2, 133 - 150

OPTIMIZING DISTRIBUTION NETWORKS WITH NESTED GENETIC ALGORITHMS

O. BRUDARU and B. VALMAR

Abstract. This paper deals with the optimizing of distribution networks with a central depots and a prescribed number of intermediate depots that supply groups of clients. A two levels metaheuristic is described for solving it. On the first level, a genetic algorithm used for finding the feasible group of consumers and the corresponding intermediate depots, like in the *p*-median problem. For such a partitioning of the clients, the interior provisioning circuits are obtained by invoking a hybrid genetic algorithm, and this task represents the second level of the metaheuristic. This second level completes the partial solutions from the first level and computes the fitness function of the genetic algorithm on the first level. The performance of the metaheuristic containing the two nested genetic algorithms is experimentally evaluated.

I. INTRODUCTION

This paper proposes a new solving technique for optimizing distribution networks (ODN). In this problem, a central depot and a set of consumers are given. The set of consumers is to be partitioned into a given number of groups. Within each group, a consumer receives its demand from an intermediate depot. The intermediate depots are selected from a given subset of consumer's locations. The sum of demands of consumers forming a group does not exceed the capacity of the vehicle assigned to that group. Within each group, the transportation is done by using a Hamiltonian circuit passing trough all the members of the group and starting/ending from/at the respective intermediate depots.

Keywords and phrases: genetic algorithms, metaheuristics, networks
(2000) Mathematical Subject Classification: 68T20, 90-08

There are two possibilities to supply the intermediate depots from a central depot: independently, using several transportation means or using a single vehicle that passes successively through each intermediate depot. The transportation cost depends on the carried load and the unitary costs that are associated with the connections between depots/consumers. The total cost associated to a partitioning of consumers is the sum of all intermediate costs, including the cost for provisioning the intermediate depots. The problem is to find a partitioning of the consumers and the corresponding supplying routings so that the total transportation cost is minimized.

In [23] it is studied a variant of the vehicle routing problem with time window constraints and the vehicle capacity constraints for which a two stage algorithm is proposed. In the first stage the algorithm maximizes the number of customers served using an ejection pool for temporarily storing the clients that are not (yet) supplied. During the second stages, one attempts to minimize the total travel distance using a hill-climbing algorithm. The algorithms used in [36] for vehicle routing problem are artificial intelligent algorithms such as simulating annealing and tabu search. It is proposed a new cooling scheme leading to a first simulated annealing application for vehicle routing problem, and tabu search with both recency and frequency measures. In [35] a comprehensive empirical study is presented on the effects of genetic operations on the population diversity of the genetic algorithms solving vehicle routing problem and a universal adaptive control function is proposed to maintain the population diversity. A genetic algorithm [24] for vehicle routing is presented in [32]. It uses a global customer clustering method that uses an adaptive search strategy to assign vehicles customers. A local post-optimization method is applied to produce results superior to those obtained by competing heuristic search methods. In [33] a hybrid search strategy that combines genetic algorithms, simulates annealing and tabu search for vehicle routing with time windows is described. In this hybrid, a global search made by genetic algorithm produces an initial solution to the problem. This solution is improved successively by a customer interchange method guided by tabu search combined with simulated annealing that are known as efficient local neighborhood strategies.

A two levels metaheuristic is described for solving it. On the first level, a genetic algorithm used for finding the feasible group of consumers and the corresponding intermediate depots, like in the *p*-median problem. For such a partitioning of the clients, the interior provisioning circuits are obtained by invoking a hybrid genetic algorithm, and this task represents the second level of the metaheuristic. This second level completes the partial solutions from the first level and computes the fitness function of the genetic algorithm on the first level. The performance of the metaheuristic containing the two nested genetic algorithms is experimentally evaluated.

Section 2 describes a mathematical model of the problem. The subproblems invoked in the solving of ODN problem are described and their solving techniques are outlined in section 3. The metaheuristic for solving the main variant of ODN problem is described in section 4. The results of the experimental evaluation of the metaheuristic are presented in section 5. Last section summarizes the work.

II. PROBLEM FORMULATION

Consider a set of n consumers denoted by $\{1,2,...,n\}$ and a central depot named "0". Let $G=(V,A)$ be an oriented graph where $V=\{0,1,2,...,n\}$ and the set of arcs A containing all pairs (u,v), $u,v \in V$ for which u is directly connected to v. Suppose that $V' \subseteq V_c$ is the set of locations that can become intermediate depots and denote by p the prescribed number of intermediate depots. Denote by $d(i)$ the demand of consumer i, $i=1,...,n$.

Let m be the number of available vehicles and denote by $c(1),...,c(m)$ their corresponding capacities. Consider that $P=\{P_1,...,P_n\}$ is a partition of $V \setminus \{0\}$ and $g:\{1,...,p\} \to \{1,...,m\}$ is the function that associates the vehicle $g(i)$ with group P_i, $i=1,...,p$. Also, consider the injective function $r:\{1,...,p\} \to V'$, where $r(i)$ is the intermediate depot associated with group P_i, $r(i) \in P_i$, $i=1,...,p$.

The first condition requires that

$$D(P_j)=\sum_{h \in P_j} d(h) \le c(g(j)),\ \ j=1,...,p. \tag{1}$$

As in section 4.1, denote by $\cos t(i,j)$ the unitary cost of transportation from i to j, $i,j \in V$. Denote by $\delta(h)$ the weight of the empty vehicle h, $h=1,...,m$. Similarly, the cost of transporting load q from i to j with vehicle h is $c(q;i,j)=(q+\delta(h))\cdot \mathrm{cost}(i,j)$.

Let $C(P_j;g(j),r(j),\pi_j)$ be the minimum cost of provisioning group P_j, with truck $g(j)$, starting from $r(j)$ and returning to $r(j)$ on a Hamiltonian circuit π_j passing through P_j like in the model of TDT problem formulated in section 4.1. During this transportation, the consumers in $P_j \setminus \{r(j)\}$ are provisioned, whilst the consumer corresponding to $r(j)$ is supplied in a previous stage. These specifications are illustrated in Fig. 1 (for $p=3$).

Two options are available for carrying the cumulated demands $D(P_1),...,D(P_p)$ from "0" to the intermediate depots $r(1),...,r(p)$.

Radial connecting network. The local depots are independently supplied with p vehicles $T(1),...,T(p)$, having their own weight $R(1),...,R(p)$, respectively.

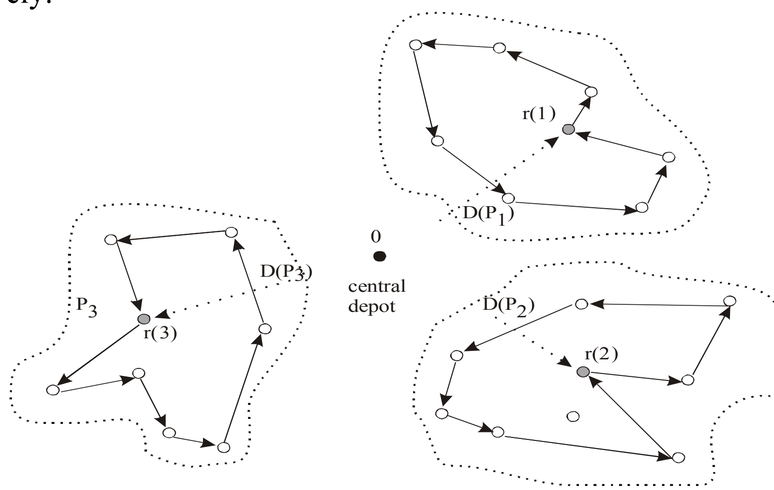

Fig. 1. Distribution network (p=3)

The vehicle assigned to group P_j has enough capacity to transport the entire demand $D(P_j)$ of the whole group P_j at once. This scheme is shown in Fig. 2.

The cost to carry the loads from "0" to the intermediate depots is given by

$$TI^{(1)}(P;g,r)=\sum_{j=1}^{p}\left[\left(D(P_j)+R(j)\right)\cdot \text{cost}\left(0,r(j)\right)+R(j)\cdot \text{cost}\left(r(j),0\right)\right] \qquad (2)$$

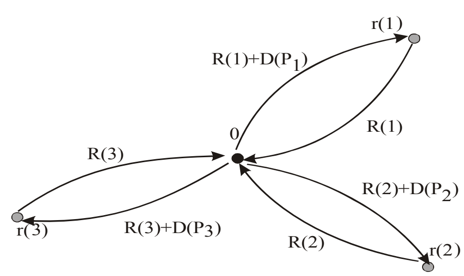

Fig. 2. Reaching the intermediate depots $r(1)$, $r(2)$ $r(3)$ by a radial network ($p=3$).

Circular connecting network. This scheme considers a permutation $\alpha = \{\alpha_1, ..., \alpha_p\}$ of the intermediate depots indicating to reach them in the order $r(\alpha_1), ..., r(\alpha_p)$. Let D be the total demand $D = d(1) + ... + d(n)$ and denote by R the own weight of the vehicle transporting the load D. The cost to supply the intermediate depots is

$$TI^{(2)}(P; g, r) = \sum_{j=1}^{p} \text{cost}\left(r(\alpha_{j-1}), r(\alpha_j)\right) \cdot \left[R + D - \sum_{h=1}^{j-1} D(r(\alpha_h))\right] \quad (3)$$

where $r(\alpha_0) = r(\alpha_{p+1}) = 0$.

This scheme is illustrated for $p = 3$ in Fig. 3, where the order to reach the depots is $r(3)$, $r(1)$, $r(2)$.

The problem of optimizing the distribution networks (ODN) has two variants:

ODN1: Find a partition $\{P_1, ..., P_p\}$ of $V \setminus \{0\}$, an assignment g of vehicles, an assignment r of intermediate depots and the routing π_j within group P_j, $j = 1, ..., p$ so that the total cost

$$T^{(1)}(P; g, r, \pi) = TI^{(1)} + \sum_{j=1}^{p} C\left(P_j; g(j), r(j), \pi_j\right) \quad (4)$$

is minimized.

ODN2: Find a partition $\{P_1, ..., P_p\}$ of $V \setminus \{0\}$, an assignment g of vehicles, an assignment r of intermediate depots, the routing π_j of group P_j, $j = 1, ..., p$ and a routing of the intermediate depots α so that

$$T^{(2)}(P; g, r, \pi, \alpha) = TI^{(2)} + \sum_{j=1}^{p} C\left(P_j; g(j), r(j), \pi_j\right) \quad (5)$$

is minimized.

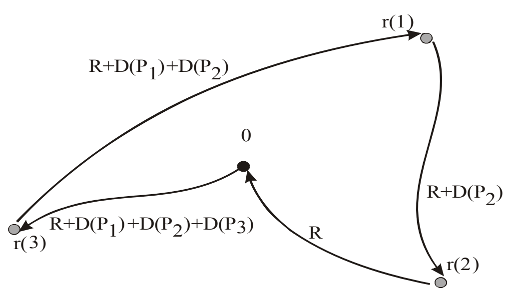

Fig. 3. Circular network like in TDT problem for supplying intermediate depots ($p = 3$, $\alpha = (r(3), r(1), r(2))$.

Each variant can be extended by supposing that each consumer has an offer $\overline{d}(i)$, $i = 1, ..., n$. When a vehicle reaches consumer i, it unloads the cargo

$d(i)$ and loads $\bar{d}(i)$. So, the current load diminishes with $d(i)$ and increases with $\bar{d}(i)$ and the resulted values must be supported by the capacity of that vehicle. Since in practice the introducing of such extension do not create unfeasible solutions because usually $\bar{d}(i) < d(i)$, this extension is maintained only for the sake of completeness. The proposed method and its implementation actually ignore this extension.

It is worth mentioning that the classical vehicle routing problem is a particular case of ODN problem in the sense that the first is obtained from ODN problem by taking $p = 1$, $r(1) = 0$ and the cost of moving the load q from i to j simply equal to $\mathrm{cost}(i, j)$ (the transportation cost does not depend on the useful or residual transported load).

III. TWO SUBPROBLEMS

In this section two important subproblems appearing in the solving of ODN problem are outlined, namely *district partitioning*(p-median) *problem* and *time dependent transportation* (TDT) problem.

3.1. District partitioning problem

District partitioning problem considers a set of facility locations from which one selects *p* special locations called *centers*. Around these centers one constitutes a given number of groups. Each location has its own weight whilst a cost is associated with a direct connection from a location to another. The cost associated with a center is the sum of costs of transferring the weight of a member of the group to its center, i.e. that weight multiplied by the cost of the connection from the facility to the center. The total cost is the sum of costs of all centers. If the group size is not restricted then the problem is called *uncapacited*. The problem is to find the centers and the groups built around them so that the total cost is minimized. It is known as p-median problem, where p is the prescribed member of groups.

As shown in [21], this optimization problem is NP-hard for two or more dimensions, when either the Euclidian or the rectilinear distance measure is used and it is polynomialy solvable in one dimension. As mentioned in [29], the problem is NP-hard on general graphs with arbitrary p. In [18] and [22] it is stated that polynomial time algorithms exist for arbitrary p and the graph is a tree or a general network if p is fixed. Nevertheless, for fixed p the problem is not computationally easy.

A lot of techniques [29] in the class of local search algorithms are given in the literature: greedy local search [31], [34], [28], tabu search [30], simulated annealing [10], variable neighborhood search meta-heuristic in [13] and [19], a greedy randomized adaptive search procedure [27], boolean programming approaches in [2] and [1], genetic algorithms [16], [20], [25], [14], [17], [11], [4]. A hybrid evolutionary algorithm is described in [3]. In [7] a new hybrid genetic algorithm is proposed. It combines the approach in [12] with heuristic method [26] having good reported performance.

In the ODN problem the role of p-median subproblem is to efficiently construct the set of p centers and the groups of clients assigned to the candidates. The description of the hybrid genetic algorithm that evolves the set of *p* centers and the corresponding partitioning can be found in [7]. The distinction consist in the fact that the fitness function is mainly valued by the costs of transportation of *p* circuits, which are delivered by solving *p* instances of the transportation problem of the type described in the next section.

3.2. Time dependent transportation problem

In the time dependent transportation problem, a vehicle has to supply a set of consumers with known demands. It starts from a unique warehouse load with the whole demand, passes exactly once through each client delivering the corresponding demand and returns at the starting point. The transportation cost between two locations depends on the unitary cost between them and the weight of the transported load plus the weight of the empty vehicle. The problem is to find a Hamiltonian circuit that minimizes the total transportation cost. This problem generalizes the well-known traveling salesman and it is on its turn NP-hard. Since the cost of traversing the edge $e = (i, j)$ depends on its position in the path this problems will be called *time-dependent transportation* (TDT) problem. This problem presents its own interest but it is contained as a critical subproblem in different variants of the vehicle routing problem [15]. In the second case, an efficient way to solve this problem is very important especially when it is invoked many times for computing the fitness function of a genetic algorithm that is used for solving the main problem. A pure genetic algorithm for this problem is described in [9]. In [6] it the grafting of a heuristic method inspired by a pseudo-dynamic programming approach [5] and a genetic algorithm are described. The hybrid method resulted by combining a genetic algorithm and a branch and bound method for solving TDT is described in [8]. This hybrid genetic algorithm is used in the present approach for computing the cost of a partitioning coming from the p-median subproblem.

4. METAHEURISTIC FOR RADIAL DISTRIBUTION NETWORK

In this section metaheuristic method is described proposed to solve the radial provisioning of the intermediate centers. Since the first/second level of metaheuristic is basically very similar to the hybrid genetic algorithm for p-median and TDT problem, respectively, only the specific differences are discussed.

4.1. Metaheuristic's first level

Its main function is to establish the intermediate depots that correspond to centers in p-median problem and to establish the group of consumers around them. The structure of group P_j is described by the list $\lambda_j = (r(j); v_1^j, ..., v_{n_j}^j)$, where $P_j = \{r(j); v_1^j, ..., v_{n_j}^j\}$ and n_j is the number of consumers around the intermediate depots $r(j)$. For ODN problem the members of P_j must the specified because this time their order is important and is obtained by solving an instance of TDT problem. A chromosome is essentially the concatenation $\lambda = \lambda_1 \circ ... \circ \lambda_p$ of the list describing the clusters of clients. The individuals for the initial population are generated like for p-median problem by selecting randomly the intermediate depots among the elements of set V'. This function r is defined for a new individual.

In order to define function g for a new individual, the following rule is applied. It is supposed that the available vehicles are sorted in descending order of their capacity and the total available capacity exceeds the total demand. From this list, one takes the leftmost unassigned vehicle v, randomly select an intermediate depot $r(j)$ that does not have a vehicle, and assign v to $r(j)$. Therefore, both function r and g are defined for current chromosome.

Then, a feasible assignment of consumers to the depots $r(1), ..., r(p)$ is accomplished by applying the following procedure:

Heuristic for client assignment

0. initialize $P_j = \varnothing$, $j = 1, ..., p$;

i. for each consumer $i \in V \setminus \{r(1), ..., r(p)\}$ do:

i.i. construct the permutation $\xi = (\xi_1, ..., \xi_p)$ of intermediate depots so that
$\text{cost}(\xi_1, i) \le \text{cost}(\xi_2, i) \le ... \le \text{cost}(\xi_p, i)$;

i.ii. determine the smallest $j = \{1, ..., p\}$ for which

$\sum_{h \in P_j} d(h) + d(i) \le c(g(j))$ and add i to P_j.

The above heuristic is very similar to the rule that constructs a solution of the p-median problem for a given set of centers. Step (ii) ensures that the capacity of a vehicle serving a certain depot is not exceeded.

If a circular scheme is used to supply the intermediate depots, the permutation α of depots is added as prefix to the current chromosome. This part of the chromosome requires a special treatment. As soon as groups $P_1,...,P_p$ are obtained, the subproblem of finding a good order to supply the groups for minimizing $TI^{(2)}$ defined by (3) is an instance of TDT. This can be easily solved by invoking the hybrid genetic algorithm because $p << n$. For small values of p ($2 \le p \le 5$), $TI^{(2)}$ can be minimized by enumerating all the permutations of $\{1,..., p\}$. Therefore, no major differences between the variants of ODN problem exist.

It is worth mentioning that the including of the whole structure of groups in the chromosome structure is motivated only by the use of some genetics operators that operate small changes between groups in the final stages of the evolution. Experiments have shown that, actually adopting a shorter chromosome and a larger population size can compensate the lack of this type of fine tuning. Further, the chromosome contains only the intermediate depots and information about g function. So, the genetic algorithm implementing the first level of the metaheuristic remains a constructive one.

The best found routing solutions corresponding to the groups of clients are temporarily stored in a file only for the new chromosomes produced in current evolution stage. These solutions are permanently stored only for the best solution. This prevents the eventual difficulty to find the best routing solutions again for the best individuals.

The storing of the routing in the groups forming the solution could be avoided by computing of the fitness of the best chromosome again but sometimes these solutions to TDT cannot be found again because the hybrid genetic algorithm for TDT is not deterministic.

4.2. Metaheuristic's second level

For a given chromosome, one obtains the groups of clusters $P_1,...,P_p$ constituted around the depots $r(1),...,r(p)$, respectively, by using the heuristic for client assignment described above. For each $j \in \{1,..., p\}$, P_j represents an instance of TDT in which the role of "0" is played by the depot $r(j)$. Whilst the first part of the total cost in (1) or (2) can be computed with the available information, the second part in (1) and (2) does not. This task is accomplished by the hybrid genetic algorithm for solving TDT described in [8]. The better this algorithm works the more precise is the returned value of (2) or (3) is. The value of (2) or (3) is the fitness function value for the current chromosome. For each chromosome, p TDT instances are to be solved with the hybrid genetic algorithm.

Experiments show that the exact values of the fitness function are not really necessary, but a correct enough hierarchy of individuals with regard the their performance does. Consequently, in order to reduce the computing time it is enough to let the hybrid genetic algorithm for TDT work with no more than 60% of the number of iterations needed to obtain the most accurate results.

V. EXPERIMENTAL RESULTS

The first objective of the experimental investigation was to test the ability of the metaheuristic for finding known optimal solutions. A set of artificial instances like that depicted in Fig. 4 was produced. The demands of consumers in each cluster decreases along the current circuit shown by circular arrow within each group. The unitary cost are proportional to Euclidian distance between two points, whilst

$\mathrm{cost}(0,r(1)) < \mathrm{cost}(0,r(2)) < \ldots < \mathrm{cost}(0,r(p))$ and $D(r(1)) > D(r(2)) > \ldots > D(r(p))$.

Each circuit is formed by the points of convex polygon with equal sides. Only the optimal solution is drawn in Fig. 4.

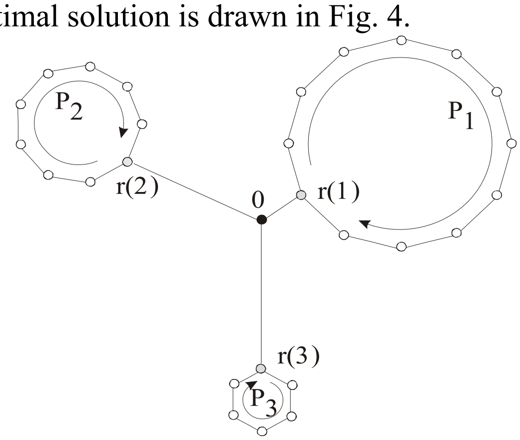

Fig. 4. Example of artificial with known optimal solution.

A set of 10 regular structures with random values of p between 2 and 10 was generated, each group having a random number of clients in the range

4-20. The demands within each group vary in arithmetic progression. The starting values are greater as the distance from the corresponding intermediate points to "0" is smaller. For each such structure a number of 20 runs were executed and for each run the ratio $T^* / T^{(1)}$ was computed, where T^* is the optimal cost of the corresponding instance and $T^{(1)}$ is defined by (4). The obtained distribution is shown in Tab. 1.

Tab 1. Distribution, average and standard deviation of $T^*/T^{(1)}$

$T^*/T^{(1)}$	<0.82	[0.82,0.88)	[0.88,0.92)	[0.92,0.96)	[0.96,1]
rel. freq.	0.0298507	0.0656716	0.1283582	0.2567164	0.519403
average=9	0.94620	std.dev.=36	0.17923		

The resulted average is 0.946. The graph of this distribution is shown in Fig. 5. The conclusion is that the method can detect the optimal solutions efficiently.

The second objective is to get an idea on the stability of the method. The value of the standard deviation obtained for the above mentioned sample is 0.179 and this shows that the method gives good results systematically.

Concerning the computational effort, the average of the number of iterations made until the step condition became true is up to 1200, whilst the required computing time was about 4 minutes for ACPI Multiprocessor PC with two Intel (R) Core (TM) 2 Duo CPU T7250 @ 2.00 GHz.

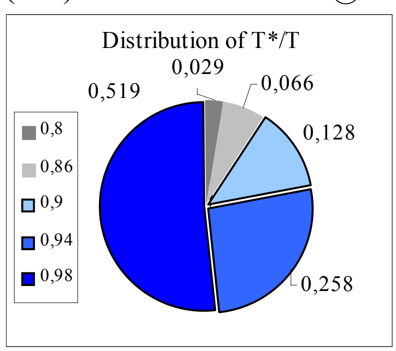

Fig. 5. Distribution of T*/T

The second experiment focuses on the evaluating the performance when the metaheuristic acts on some benchmarks dedicated to vehicle routing problems and adapted for ODN problem.

The instances have been downloaded from the address [37]. These VRP instances supported the following adaptations:

(i) the coordinates of the locations were used to compute unitary cost proportional to the Euclidian distances;
(ii) the weight of the empty vehicle represents about 60%-80% of the total demand divided by p;
(iii) the demand values are the original ones;
(iv) the number of vehicles equals $p+2$.

Three instances with $n=36$ and $p=5$, $n=53$ with $p=7$ and $n=70$ with $p=10$ were selected. Population size was set to 60 individuals. The mutation and crossover probabilities were 0.1 and 0.5 respectively. As stop condition the stabilization of the poorest performance individual for three successive generations was used. Ten runs were executed for each instance. The results are summarized in Tab. 2-4.

The last two columns in these tables represent the values of best fitness value obtained in 10 runs ($c_{\min}$) divided by the value of the best fitness obtained in the current run (c_{crt}). The average value of this ratio is about 0.86 for the instance with 36 consumers, whilst the standard deviation has a small value, namely about 0.072. Small values of the standard deviation of this indicator were obtained for all tested instances. This proves a very good stability of the method.

As can be seen in Tab. 3, for $n=53$ and $p=7$, the average of $c_{\min}/c_{crt}$ is 0.819 whilst the corresponding standard deviation is 0,093. In this case, the final value of the population size was set to 60 adult chromosomes. The results in Tab. 4 correspond to a final population of 80 individuals. The standard deviation is a bit higher than the value obtained for $n=36$ and $p=5$. In the last example, the stochastic feature of the metaheuristic is more evident as can be seen from the higher value of standard deviation, namely 0.104.

Tab. 2. Experimental results for $n=36$ and $p=5$.

run	**current cost** c_{crt}	**# iter.**	**time (sec)**	**time/iter.**	$c_{\min}/c_{crt}$
1	68149	500	580	1.160	0.830
2	65352	750	1000	1.333	0.865
3	71667	540	610	1.129	0.789
4	65474	1050	1020	0.971	0.864
5	71065	290	720	2.482	0.796
6	69882	190	390	2.052	0.809
7	66163	360	610	1.694	0.855
8	56586	420	1120	2.666	1.000
9	56929	340	970	2.852	0.993
10	69287	390	1410	3.615	0.816

average	66055.4	483	843	1.995	0.862
$c_{\min}$	56586				
$c_{\max}$	71667				
std.dev.					0.072

Tab. 3. Experimental results for $n = 53$ and $p = 7$.

run	current cost c_{crt}	# iter.	time (sec)	time/iter.	$c_{\min} / c_{crt}$
1	92130	310	1450	4.677	0.769
2	97620	350	1790	5.114	0.726
3	72538	360	1550	4.305	0.977
4	95599	420	1890	4.500	0.741
5	70927	590	1720	2.915	1.000
6	80875	370	2150	5.810	0.876
7	89540	380	1730	4.552	0.792
8	93594	420	2140	5.095	0.757
9	93000	1220	3360	2.754	0.762
10	89388	260	1490	5.730	0.793
average	87521.1	468	1927	4.545	0.819
$c_{\min}$	70927				
$c_{\max}$	97620				
std.dev.					0.093

Tab. 4. Experimental results for $n = 70$ and $p = 10$.

run	current cost c_{crt}	# iter.	time (sec)	time/iter.	$c_{\min} / c_{crt}$
1	337299	220	940	4.272	0.703
2	376041	320	1070	3.343	0.631
3	286636	230	960	4.173	0.8276
4	272404	370	1300	3.513	0.871
5	333360	390	1660	4.256	0.711
6	282451	490	1200	2.448	0.8408
7	291123	490	1890	3.857	0.815
8	306859	400	1460	3.65	0.773
9	237315	440	2150	4.886	1.000
10	355237	440	1440	3.272	0.668
average	307872.5	379	1407	3.767	0.784
$c_{\min}$	237315				
$c_{\max}$	376041				
std.dev.					0.104

Fig. 6 shows the variation of standard deviation of the relative performance indicator as a function of the number of consumers. The trendline indicates a very good resemblance with a logarithmic model.

As a conclusion, the proposed metaheuristic incorporating two genetic algorithms, one for p-median and other for the time dependent transportation problem has good performance, even in the actual hypostasis where it does not incorporate the best variants of these basic components. Moreover, it constitutes a good objective of a distributed implementation. These extensions are left for future investigations.

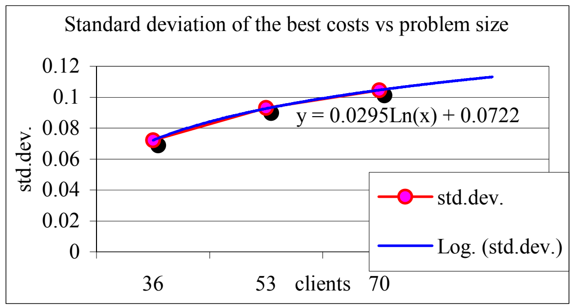

Fig. 6. Standard deviation of best cost indicator as a function of the number of consumers.

VI. CONCLUSIONS AND FURTHER RESEARCH

This chapter presented a new metaheuristic for optimizing distribution networks.

A distributed implementation of the time intensive metaheuristic for optimizing distribution network, is of great interest because the high level genetic algorithm forming this method invokes the genetic algorithm for TDT problem many times for each new candidate partition. Extending this segregative genetic algorithm to pickup-and-delivery problem is of real interest. Also, grafting appropriate heuristics for TSP on this segregative genetic algorithm could transform it into a new stronger solving tool for this particular problem.

References

[1] Avella, P., Sassano, A., Vasilev, I., **Computational study of large-scale p-median problems**, Math. Program., Ser. A 109, pp. 89-114, 2007.

[2] Beltran, C., Tadonki, C., Vial, J.-Ph., **Solving the p-Median Problem with a Semi-Lagrangian Relaxation**, September 21, 2005.

[3] Borgulya, I., **A hybrid evolutionary algorithm for the p-median problem**, GECCO'05, June 25-29, Washington, DC, USA, 2005.
[4] Bozkaya, B., Zhang, J., and Erkut, E., **An efficient genetic algorithm for the p-median problem**, Z. Drezner and H. Hamacher editors, Facility Location: Applications and Theory, pp. 179–205. Berlin: Springer, 2002.
[5] Brudaru, O., **An Efficient Heuristic Method for the Finding of a Minimal Hamiltonian Circuit in Connection with a Transport Problem**, Bul. I. P. Iasi, tom XXVII (XXXI) f. 1-2, s1 Matematica, 1981, pp. 61-64.
[6] Brudaru, O. and Vîlcu, A., **Genetic Algorithm with Accelerating Hybrid Components for Affine Cost Hamiltonian Circuits**, ICPR-16, 16th International Conference on Production Research, 30 July-3 August, 2001, Prague.
[7] Brudaru, O., Valmar, B., **A Hybrid Genetic Algorithm for p-median problem**, 8th International 2004, Research/Expert Conference: Trends in the Development of Machinery and Associated Technology, TMT Neum, Bosnia and Herzegovina, 15-19 sept. 2004, pp.875-878.
[8] Brudaru, O., Valmar, B., **Hybrid genetic-algorithm / branch & bound technique to solve a time-dependent transportation problem**, Proc. of 6th Eurosim Congress on Modelling and Simulation, B. Zupancic, R. Karba, S. Blazic (eds.) Ljubljana, Slovenia, Sept. 9-13, 2007 , vol. 2, pp. 1-7
[9] Brudaru, O., Vîlcu, A., **Genetic algorithm for a transportation problem with variable load along hamiltonian circuits**, 10th International DAAAM Symposium "Intelligent Manufacturing & Automation: Past - Present - Future", Oct. 21-23 1999, Vienna, Austria.
[10] Chiyoshi, F., Galvano, D., **A statistical analysis of simulated annealing applied to the p-median problems**, Annals of Operations Research, 96, pp. 61-74, 2000.
[11] Chiou, Y. and Lan, L. W., **Genetic clustering algorithms**, European Journal of Operational, Research, 135(2):413–427, 2001.
[12] Correa, E. S., Steiner, M.T.A., Freitas, A.A., Carnieri, C., **A genetic algorithm for solving a capacitated p-median problem**, LE Spector and E Goodman et al, editors, Morgan Kaufmann, Procedings of 2001 Genetic and Evolutionary Computation Conference (GECCO-2001), pp. 1268-1275, San Fracisco, USA, July, 2001.
[13] Daskin, M. S., **Network and discrete location: models, algorithm and application**, John Wiley and Sons, Inc., NY, 1995.
[14] Dibble, C., Densham, P.J., **Generating interesting alternatives in GIS and SDSS using genetic algorithms**, GIS/LIS symposium, University of Nebraska, Lincoln, 1993.

[15] Domschke, W., **Logistik: Rundreisen und Touren (Bd.2)**, 2, Aufl., Oldenbourg, Munchen, 1989

[16] Dvorett, J., **Compatibility-based genetic algorithm: A new aproach to the p-median problem**, Department of Industrial Engineering and Management Sciences Northwestern University Evanston, IL 60208, June 1999.

[17] Erkut, E., Bozkaya, B., Zhang, J., **An effective genetic algorithm for the p-median problem**, (In Press.), 2001.

[18] Garey, M.R., Johnson, D.S., **Computer and Intractability: A guide to the theory of NP-completeness**, W.H. Freeman & Co., San Francisco, 1979.

[19] Hansen, P., Jaumard, B., **Cluster analysis and mathematical programming, Mathematical Programming**, 79, pp. 191-215, 1997.

[20] Hosage, C.M., Goodchild, M.F., **Discrete space location-allocation solution from genetic algorithms**, Annals of Operational Research, 6, pp. 35-46, 1986.

[21] Jakson, L. E., Rouskas, G. N., Stallman M. F.M., **The directional p-median problem with applications to traffic quantization and multiprocessor scheduling,** Jackson L. (ed), Rouskas G.Doctoral Thesis, http://portal.acm.org/citation.cfm?id=1023151.

[22] Kariv, O., Hakimi, S. L., **An algorithmic approach to network location problem-II, The p-medians**, SIAM Journal on Applied Mathematics 37(3), pp. 539-560, 1979.

[23] Lim, A., Zang, X., **A Two-Stage Heuristic for the Vehicle Routing Problem with Time Windows and a Limited Number of Vehicles**, Proc. Of the 38th Hawaii Intern. Conf. On System Sciences, 2005.

[24] Michalewicz, Z. **Genetic Algorithms + Data Structures = Evolution Program**, 2nd ed., Springer Verlag, Berl,in, 1994.

[25] Moreno-Perez, J.A., Moreno-Vega, J.M., **Annealing in p-median problems**, Talk at the Canadian Operational Research Society Conference, Montreal, 1994.

[26] Pizzolato, Nelio Domingues, **A heuristic for large-size p-median location problems with application to school location**, Annals of Op. Research vol.50 , pp. 473-485, 1994

[27] Resende, M. G. C., Werneck R.. F., **A GRASP with path-relinking for the p-median problem**, Sept., 2002.

[28] Resende M.G.C., Werneck, R.F., **On the implementation of a swap-based local search procedure for the p-median problems**, In: Ladner, R.E. (ed.), Proceedings of the 5th Workshop on Algorithm Engineering and Experiments (ALENEX'03), pp. 119-127, 2003.

[29] Reese, J., **Methods for Solving the p-Median Problem: An Annotated Bibliography**,http://ramanujan.math.trinity.edu/tumath/research/reports/report96.pdf, accessed Feb., 2009

[30] Rolland, E., Schilling, D.A., Current, J.R., **An efficient tabu search procedure for the p-median problems**, EJOR, 96, pp. 329-342, 1996.

[31] Teitz, M.B., Bart, P., **Heuristic methods for estimating the generalized vertex median of a weighted graph**, Operations Research, 16, pp. 955-961, 1968.

[32] Thangiah, S.R., **Vehicle Routing Problem with Time Windows using Genetic Algorithms**, Practical Handbook of Genetic Algorithms - New Frontiers, vol. II, L. Chambers (ed.), CRC Press, Boca Raton, pp. 253.

[33] Thangiah, S.R., **A Hybrid Genetic Algorithms, Simulated Amealing Tabu Search Heuristic for Vehicle Routing Problem with Time Windows**, Practical Handbook of Genetic Algorithms - Complex Coding Systems, vol. III, L. Chambers (ed.), CRC Press, Boca Raton, pp. 347.

[34] Whitaker, R.A., **A fast algorithm for the greedy interchange for large-scale clustering and median location problems**, INFORS, 21, pp. 95-108, 1983.

[35] Zhu, K.Q., **Population Diversity in Genetic Algorithm for Vehicle Routing Problem with Time Windows**, Dept. of Comp. Scy., National University of Singapore, 2004.

[36] Zhu, K.Q., Tan, K.C., Lee, L.H., **Heuristic for Vehicle Routing Problem with Time Windows**, National University of Singapore, 1999.

[37] http://neo.lcc.uma.es/radi-aeb/WebVRP/data/instances/Augerat/A-VRP.zip, accessed Jan. 2007.

O. Brudaru[*, **], B. Valmar[**]

[*]Institute of Computer Science, Romanian Academy, Iaşi Subsidiary,

[**] "Gh. Asachi" Technical University Iaşi, Department of Management and Production Systems Engineering, Iasi, Romania

e-mail: brudaru@tuiasi.ro