Int. Journal of Business Science and Applied Management, Volume 7, Issue 3, 2012

An evolutionary algorithm for a real vehicle routing problem

Panagiotis Adamidis
Alexander TEI of Thessaloniki, Department of Information Technology
P.O. BOX 141, 57400 Sindos Thessaloniki, Greece
Telephone: +30 2310791285
Email: adamidis@it.teithe.gr

Christos Voliotis
Alexander TEI of Thessaloniki, Department of Information Technology
P.O. BOX 141, 57400 Sindos Thessaloniki, Greece
Telephone: +30 2310013985
Email: gchvol@gmail.com

Evaggelia Pliatsika
Alexander TEI of Thessaloniki, Department of Information Technology
P.O. BOX 141, 57400 Sindos Thessaloniki, Greece
Telephone: +30 2310013985
Email: evaggeliapliatsika@gmail.com

Abstract

The NP-hard Vehicle Routing Problem (VRP) is central in the optimisation of distribution networks. Its main objective is to determine a set of vehicle trips of minimum total cost. The ideal schedule will efficiently exploit the company's recourses, service all customers and satisfy the given (mainly daily) constraints. There have been many attempts to solve this problem with conventional techniques but applied to small-scale simplified problems. This is due to the complexity of the problem and the large volume of data to be processed. Evolutionary Algorithms are search and optimization techniques that are capable of confronting that kind of problems and reach a good feasible solution in a reasonable period of time. In this paper we develop an Evolutionary Algorithm in order to solve the VRP of a specific transportation company in Volos, Greece with different vehicle capacities. The algorithm has been tested with different configurations and constraints, and proved to be effective in reaching a satisfying solution for the company's needs.

Keywords: vehicle routing problem, evolutionary algorithms

1 INTRODUCTION

The Vehicle Routing Problem (VRP) is a combinatorial optimization problem trying to service a number of customers with a fleet of vehicles. It is an important problem in the fields of transportation, distribution and logistics. Laporte (1992) presents a definition of classical VRP along with some approximation algorithms to solve it.

VRP is associated with the efficient use of a company's resources and therefore constitutes an important factor in reducing costs. The challenge in such problems is finding a route schedule that satisfies all constraints at the lowest possible cost. These problems have many common and some different features as well. Common features include the existence of a vehicle fleet and a pool of customers that have to be served by the vehicles. Different features include variations of VRP with minimization of vehicle number or/and route number, different vehicles, different constraints, time windows etc.

The vehicle routing problem is a nonlinear combinatorial optimization problem with multiple constraints, and most of the proposed solutions cannot be applied effectively to real problems.

Generally, metaheuristics provide much better solutions on large-scale real instances. The reader is referred to Gendreau et al. (2008) for a survey about metaheuristics for the VRP.

In this paper we present an Evolutionary Algorithm approach to solve VRP adapted to the requirements and characteristics of a transportation company in Volos, Greece. Although we discuss a specific case study, most methods are general enough and should be applicable to modelling and solving other instances of the problem. The situation in real companies is relatively complicated and involves many different aspects of VRP.

In the following section we present a description of the company's problem along with its major characteristics and restrictions. Section 3 presents some interesting approaches to solve VRP with Evolutionary Algorithms (EAs) and then in section 4 we propose an EA solution to the specific problem. Our EA approach has been tested with different EA configurations. Finally, in section 6, we conclude with a discussion of the results.

2 PROBLEM DESCRIPTION – CASE STUDY

As already mentioned, in this paper we consider the Vehicle Routing Problem adapted to the requirements and operational restrictions of a transportation private limited company (Ltd.) in Volos, Greece. Each shareholder owns a number of vehicles and is responsible for the damages incurred to his vehicles. Each shareholder is awarded a part of the profits according to the total route length travelled by the vehicles that he owns. In this case, the cost objective is not the total mileage of all the routes but rather distribute fairly every day route among all available vehicles through a period of a whole year.

As described earlier, a large number of variants of the VRP exist, adding different constraints to the problem. Our problem seems like a compilation of some VRP variants, with the main differences from classical VRP being:

- there is not a central depot but multiple pickup and delivery locations,
- vehicles have different capacities and characteristics (heterogeneous fleet VRP - HVRP),
- more than one route can be assigned to a vehicle (VRP with Multiple Use of Vehicles - VRPM),
- random demand fluctuations which is usually referred to as the stochastic or dynamic routing problem.

The vehicles are considered to have different capacities and since there is not a central depot, vehicles may be located all over Greece. A major waste of resources is when a vehicle travels empty of load (empty kilometers). This happens when the vehicle returns from a delivery location to Volos, or when going to another location to pickup goods. Therefore, one of the objectives is to decrease the distance that is covered by empty vehicles.

However, if a vehicle is at a distance of no more than 100km from the city of Volos (this distance is considered relatively small) the return is not considered harmful. In this case, the vehicle returns to Volos, unless it is scheduled for another service from current or nearby location.

Therefore the main difference of the problem to be solved lies on the objectives of the problem in hand, which are:

- fair distribution of all routes among all available vehicles. Each vehicle must service routes with total equal length in the long run, and routes of analogous difficulty.
- minimisation of empty kilometers i.e. routes covered empty of load.

Some service demands concern the port of Volos which generally result in short distance routes starting from the port and scheduled well in advance. The service of a cargo ship involves the transport of large volumes of goods and it usually lasts for many days with a large numbers of trips within each day.

Another part of the service demands involve routes starting from the industrial area of Volos as a depot, with customers to be served all over Greece.

Another category involves service demands for the transport of small volumes of goods for a few days, from different customers, from several cities to the city of Volos. These orders may be scheduled or unexpected.

Since most of the routes have Volos as the starting location, each vehicle has to return to the city of Volos after the completion of each service demand, unless it is otherwise scheduled.

There are three kinds of routes depending on origin and destination:

- Volos → city A: A vehicle located in Volos, is selected according to a virtual selection factor, described next in section 2.1.
- City A → city B: In case that there exist vehicles of the company in City A, then a vehicle is selected according to selection factor. If there are no vehicles in City A, then we choose the vehicle that is closer to City A.
- City A → Volos

Since the possible pickup locations cover all Greece, another goal is to have available vehicles to pickup locations according to customer needs. Consequently, the criteria of choosing a vehicle for a specific service demand depend on:

- the distance that will be travelled empty of load,
- service demands in relation to available nearby vehicles,
- a specifically devised virtual selection function, explained in the following subsection.

2.1 Virtual Selection Factor

In order to distribute fairly the profits among the company's shareholders, the cost objective is not the total mileage of all routes, but rather to distribute every day routes among all available vehicles through a period of a whole year.

All vehicles should cover routes of analogous difficulty (special routes) and length, at the same time. Thus we devised the concept of Virtual Selection Factor (VSF), a factor that computes a virtual route length that takes care of operational restrictions set by the company. The total mileage covered by a vehicle is of minor importance, and it is taken into consideration through VSF.

The VSF is used in selecting a vehicle for a specific route, assigning a virtual route length to each vehicle. Vehicles with low factor have more probabilities to be selected. The VSF depends on the actual route length covered by the vehicle (ARL) and the following operational restrictions:

- Load of the vehicle. A heavy load causes more damage to the vehicle. The actual route length is increased by 7% for loads over 15tn. (Ldinc)
- Routes covered by empty vehicles, either going to a pickup location or returning from a delivery location. When the route length covered with no load is over 40km, the route length is reduced by 40%. (NoLd)
- Special routes (SR). There are routes that are particularly difficult for the driver and may also cause damage to the vehicle. For such routes the actual route is increased depending on load weight. An example of such routes is given in Table 1.

Table 1: Example of special routes

Route	Empty	Light weight	Heavy weight
Brallos	15%	30%	45%
Domokos	5%	10%	15%

- Multiple delivery locations. The route length is increased by 15km for each delivery location, except from the prefectures of Attiki and Thessaloniki, where the increase is 40km per delivery location. (Mdl)
- Routes to/from islands. Route length is increased by 200km per day (Isle).

Thus the VSF is computed by equation 1.

$$VSF = ARL + Ldinc + SR + Mdl + Isle - NoLd \tag{1}$$

3 EVOLUTIONARY ALGORITHMS FOR THE VEHICLE ROUTING PROBLEM

The Evolutionary Algorithm (EA) is an adaptive heuristic search and optimization method based on population genetics. EA evolves a population of individuals, which represent problem solutions, by creating new generations of offspring through an iterative process until some convergence criteria are met. The best individual provides the corresponding solution (De Jong 2002, Eiben & Smith 2008).

Before the evolution starts we have to create the initial population choosing a suitable representation and a population initialization method. Then the evolution starts by selecting the individuals that will produce (through recombination and mutation) offspring that will form the next generation. The selection phase consists of randomly choosing two parent individuals from the population with probability proportional to their fitness, emphasizing genetic quality while maintaining genetic diversity. A new generation is created by repeating the selection, recombination, and mutation processes until a complete set of evolved new individuals have been created and placed in the new population. In some cases, all individuals in the old population are replaced by new ones, and in some cases a set of old individuals are preserved.

The general idea of using Evolutionary Algorithms and their hybrids for VRP has proven to be very efficient. In the following we present some interesting approaches.

Ozdemir and Mohaan (2000) created an application (GrEVeRT) to solve Vehicle Routing Problems with Time Windows (VRPTW). In this application the time window is fixed and cannot be violated. They used an EA with a graph incorporating all route restrictions. An individual is represented by a graph with each path composing a vehicle route. All paths represent a complete solution. Their initialization process allows for only feasible individuals to be created. The process allows for random insertion of customers into routes only if a feasible solution is created. A better version uses heuristics based on distance, service time, waiting time, and order size. They used 1 and 2 point recombination keeping restrictions under control. Finally there is a routing time-shift, whenever possible, in order to eliminate void times among routes. The objective is to minimize the total mileage of the vehicles. They also used elitism and a replacement strategy where offspring compete with their parents and the winner is inserted in the new generation.

Mak and Guo (2004) solved VRP with an Age-based GA and soft time windows applying a penalty when time window constrained is violated. A graph based representation is used and the minimization of the total mileage is the objective. The proposed EA uses an age-like feature that allows an individual to survive and reproduce for a number of generations. Individuals belong to different age groups. Extra attributes include age, birth rate and survival rate. Each individual is encoded as a character sequence with characters representing specific customers. Selection is based on the age of the individuals. Individuals in the same age group have the same possibility to be selected as parents. They used Pertialli Mapped Crossover (PMX) with correction of offspring. The mutation operator, modify individual with permutation of its contents after a random gene. They conclude that the optimal or near-optimal solutions can be obtained and this seems mainly due to its age property that allows good individuals to survive longer and gives them more chance to generate offspring.

Machado et al. (2002) implemented two EA approaches to an instance of the generic VRP. Their first approach used a standard genetic algorithm (GA) and the second a cooperative coevolutionary algorithm (CCA). The most interesting is the coevolutionary model where the problem is decomposed into components assigned to different subpopulations that are evolved simultaneously but isolated. Each individual is evaluated by selecting collaborators selected from other populations to form complete solutions. They used two subpopulations. One subpopulation described the number of nodes of each route, and the other regulated the composition of each partition and also the order by which the nodes are visited. They used PMX crossover and swap mutation. They concluded that EA techniques can deal in a satisfactory way with VRP and that the inclusion of heuristics provided significant improvement.

Jin and Hsu (2004) applied a Family Competition Genetic Algorithm (FCGA) to an instance of VRP. FCGA adapts the concept of families to traditional EAs. Once a parent is selected to mate and produce one offspring, in FCGAs this parent produces a family with other individuals. Only the best individual of the family survives keeping the population size constant. Each individual is represented as an ordered list of pickup and delivery locations. They tested four recombination operators (order-based crossover, uniform crossover, and two types of merge crossover. They conclude that FCGA improves the solutions found by a traditional genetic algorithm and that FCGA is best suited for genetic operators that explore the search space uniformly, such as uniform crossover.

Alba and Dorronsoro (2004) developed a Cellular Genetic Algorithm (cGA) where population is structured in a specified topology (neighbourhood), so that individuals may only interact with their neighbours. They used a 2D toroidal grid with size 5 as the population structure. Each individual is represented as a permutation of integers containing both customers and route delimiters. Each route is composed by the customers between two route delimiters. The population is evolved applying an edge recombination operator and three mutation operators (insertion, swap or inversion) with equal probability. They also use local optimization methods to every individual after each generation. They compared their algorithm against other (TS, GA and ant algorithms) algorithms, and conclude it is faster that the others and has also a high performance in terms of the quality of the solutions. They also conclude that the inclusion of local optimization methods, improved performance.

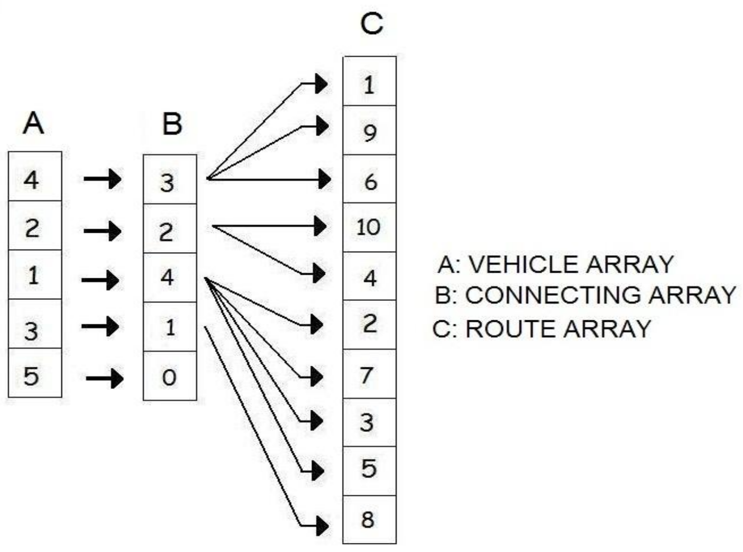
4 EVOLUTIONARY ALGORITHM SOLUTION

The creation of an EA solution involves the definition and implementation of the representation of individuals, the initialization of population, and the genetic operators (recombination and mutation) and techniques.

4.1 Representation and Initialization of Population

We started with classical representations but the final representation used in this paper is based on the problem decomposition used in coevolutionary algorithms. Each individual should have information on the available routes and the vehicles available for these routes. Thus the problem can be divided into two sub-problems, concerning either routes, or vehicles.

Figure 1: Representation of an individual



One sub-problem is route scheduling to service all customers. Each individual is represented by an array of integers. The length of the array is equal to the number of all available routes. Each array position corresponds to a specific route.

The other sub-problem is finding the appropriate vehicle for each route. Each individual is also represented by an array of integers. The length of the array is equal to the number of available vehicles. Each array position corresponds to a specific vehicle.

In order to connect these two arrays we use an auxiliary third array of integers with length equal to the vehicle array. Each cell value denotes the number of routes to be serviced by the vehicle in the corresponding position of the vehicle array. The sum of the values of this array must be equal to the length of the route array.

Routes will be serviced in the order that they appear in the route array.

Figure 1 displays an example individual where vehicle 4 will service 3 routes (1→9→6), vehicle 2 will service 2 routes (10→4), vehicle 1 will service 4 routes (2→7→3→5), vehicle 3 will service only route 8 and vehicle 5 will not move.

We used two initialization methods:

- Random Feasible Initialization (RFI). Initial population is created assigning random routes to vehicles that can service the specific routes in the specified time windows. The process follows: First, a vehicle is randomly selected and then a route. The route is assigned to this vehicle only if the vehicle can service the route in the specified time window. In this case we also update the data of this vehicle (location after service, time available for next service). The process continues for this vehicle with another random route until there are no more routes. To complete the initialization of an individual, this process is repeated for each available vehicle until all routes have been assigned to a vehicle. The above process allows for the possibility to exist unassigned orders and vehicles with no assigned routes.
- Biased Feasible Initialization (BFI). This method improves the previous one by first checking the location of the vehicle and the starting point of a route. In case they are the same, the route is assigned to this vehicle. Next, if the vehicle cannot be assigned a route starting from its location, the process continues as described in random feasible initialization.

4.2 Genetic operators and techniques

We used two different parent selection methods: Roulette wheel selection and Tournament Selection.

The recombination operator applies 1-point crossover to the vehicle array and the route array. Then, the feasibility of the child (new solution) is checked and corrected so that all routes are serviced exactly once.

Mutation is also applied to the vehicle array and the route array. We supplied two different mutation operators: swap mutation and shift mutation. In the first mutation operator the contents of two randomly chosen array positions are exchanged. In shift mutation a selected consecutive part of the array is right-sifted.

The evolution process uses generational replacement. We supplied the possibility of a competition before the child is inserted to the next generation population. A percentage of the children is not directly inserted in the next generation population. The child is first compared with the two parents and the best one is inserted to the next generation population.

Classical elitism, with the best individual of the current generation copied to the next generation, is also used.

4.3 Fitness Evaluation

The fitness evaluation has two objectives:

- minimize the total route distance covered empty of load for all vehicles.
- equalize the virtual selection factor for all vehicles.

The first objective is clearer. We compute the length of empty kilometres for all vehicles, and then we try to minimize it (Empty km).

The second objective concerns the virtual selection factor which must be equal for all vehicles. First, the VSF of each route is computed (Eq. 1). We find the maximum VSF of all vehicles, and then add the differences of the VSF of each vehicle from the maximum VSF. The objective is to minimize this sum.

The fitness evaluation function is given in equation 2.

The virtual selection factor of a vehicle is the sum of the VSF of all routes assigned to it.

$$indivFitness = w_1 * Emptykm + w_2 * \sum_{i=1}^{N} \left(VSF_{\max} - VehVSF_i \right) \qquad (2)$$

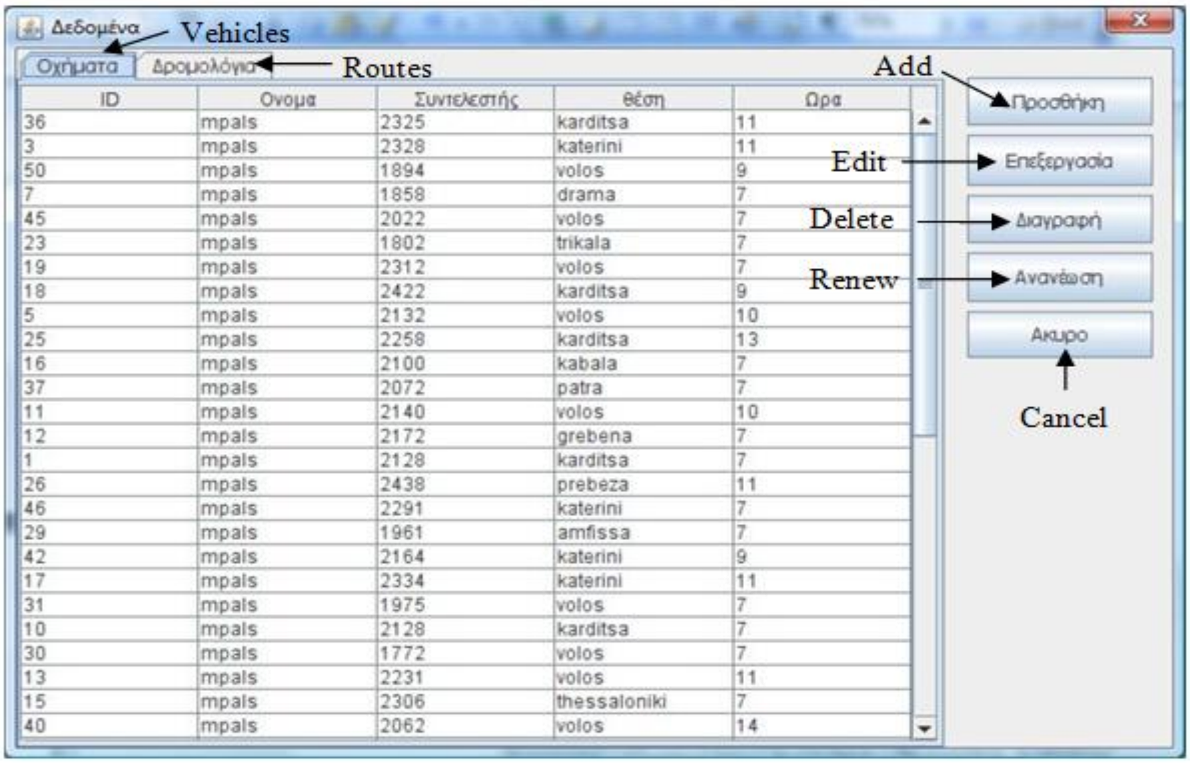
where:

Empty km is the total number of empty kilometres, N is the number of vehicles, VehVSFmax is the maximum VSF, VehVSFi is the VSF of vehicle i and w1 and w2 are the weights of each objective.

5 EXPERIMENTS AND RESULTS

We developed an application using the Java programming language to implement the EA, and PostgreSQL for the database. Greek was the language for the application interface. Figure 2, shows a screen of the application with the data processing menu.

Figure 2: Data processing menu



ID	Ονομα	Συντελεστής	θέση	Ωρα
36	mpals	2325	karditsa	11
3	mpals	2328	katerini	11
50	mpals	1894	volos	9
7	mpals	1858	drama	7
45	mpals	2022	volos	7
23	mpals	1802	trikala	7
19	mpals	2312	volos	7
18	mpals	2422	karditsa	9
5	mpals	2132	volos	10
25	mpals	2258	karditsa	13
16	mpals	2100	kabala	7
37	mpals	2072	patra	7
11	mpals	2140	volos	10
12	mpals	2172	grebena	7
1	mpals	2128	karditsa	7
26	mpals	2438	prebeza	11
46	mpals	2291	katerini	7
29	mpals	1961	amfissa	7
42	mpals	2164	katerini	9
17	mpals	2334	katerini	11
31	mpals	1975	volos	7
10	mpals	2128	karditsa	7
30	mpals	1772	volos	7
13	mpals	2231	volos	11
15	mpals	2306	thessaloniki	7
40	mpals	2062	volos	14

The experiments were set up in order to check the performance of different configurations, and the capability of the evolutionary algorithm to find feasible solutions. We used 93 different service demands, the same for every day. The 50 available vehicles are heterogeneous.

The population size of the EA was 2000 generations and the population size was 200 individuals. Elitism was always used. We run a series of 50 experiments for each configuration.

Table 2: Different EA configurations for testing

Configuration	Initialization method	Mutation – vehicle array	Mutation – route array	Replacement competition
1.1	RFI	swap	swap	0%
1.2	RFI	swap	swap	80%
1.3	RFI	shift	shift	0%
1.4	RFI	shift	shift	80%
2.1	BFI	swap	swap	0%
2.2	BFI	swap	swap	80%
2.3	BFI	shift	shift	0%
2.4	BFI	shift	shift	80%
3.1	BFI	swap	shift	50%
3.2	BFI	swap	shift	80%
3.3	BFI	swap	shift	30%
3.4	BFI	shift	swap	80%

Before running an experiment for the whole year, we compared the performance of different EA configurations. Table 2 shows the different EA configurations.

First we run experiments with configurations 1.1-2.4 in order to test the performance of the two initialization methods. Since the results (Table 3) show that BFI clearly outperforms RFI we tested some more EA configurations (3.1-3.4) with BFI.

BFI creates feasible initial solutions that take care of the service time windows to minimize empty kilometers. Starting from a better point of the search space allows for faster convergence to better solutions.

Table 3: Results from different EA configurations

Configuration	Best Fitness	Worst Fitness	Standard deviation	Empty km
1.1	183.18	1061.54	128.10	8195.2
1.2	125.89	748.69	118.83	5327.3
1.3	177.51	973.85	135.16	7887.4
1.4	165.54	700.66	103.38	7291.5
2.1	109.67	1053.05	132.50	4233.9
2.2	108.89	725.38	123.91	4281.1
2.3	108.28	954.34	135.03	4195.2
2.4	108.72	661.41	109.37	4311.5

Results from next experiment with conf. 3.1-3.4 (Table 4.) show that the EA configurations 3.1 and 3.3 outperform the others. These configurations are using shift mutation to the route array which shifts blocks of similar routes and has more possibilities in producing better feasible solutions.

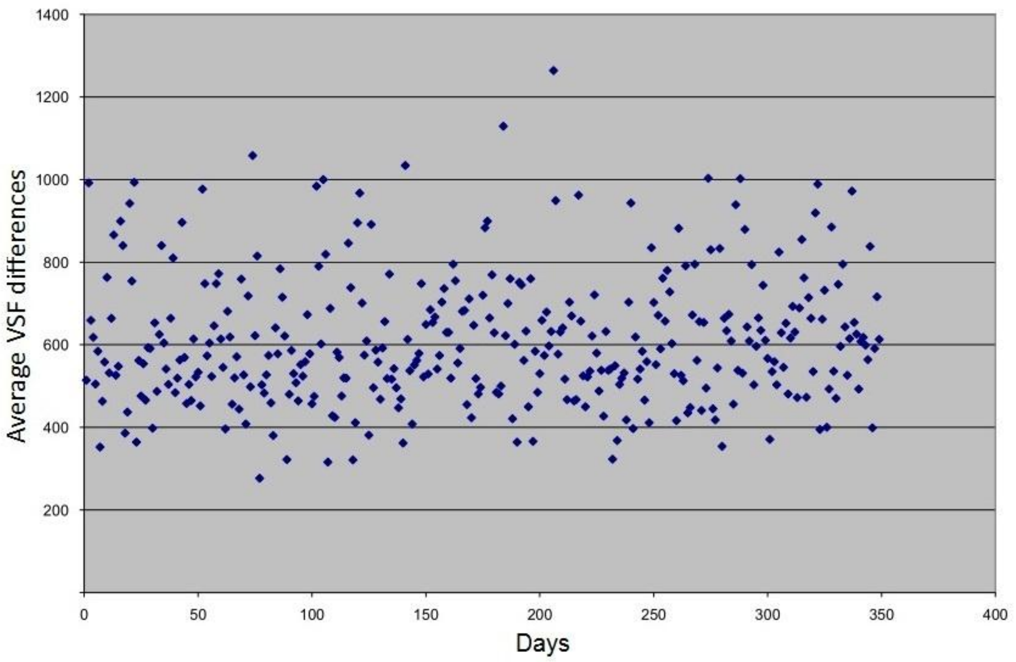
Table 4: Results from more EA configurations with BFI

Configuration	Best Fitness	Worst Fitness	Standard deviation	Empty km
3.1	107.84	779.08	117.88	4235.24
3.2	109.41	660.08	110.31	4307.12
3.3	107.84	868.71	126.22	4238.38
3.4	109.00	776.03	118.68	4272.72

Since vehicles are independent of each other, applying swap mutation on the vehicle array outperforms shift mutation in this case.

We can also conclude that the performance is improved when using competition before replacement at a rate of 30% to 50%. This is due to the longer survival of good individuals which gives them more chance to generate offspring.

Figure 3: Average Virtual Selection Function (VSF) through a year



We chose the configuration 3.1 to run a final experiment for a whole year because of the less empty kilometers. The diagram in Figure 3 proves the efficiency of the EA since the objective is to equalize the VSF of each vehicle. It is clear that the differences between VSF through a year are 300-1000km. These differences are minimal compared to the total route lengths covered by a vehicle throughout a year.

6 CONCLUSIONS

In this paper we developed an Evolutionary Algorithm (EA) in order to solve the VRP for a specific transportation company taking into consideration the company's operational restrictions.

In order to distribute fairly the profits between the company's shareholders, the most important criteria were the distance covered by empty vehicles, and the fair distribution of services among all available vehicles. The total mileage covered by a vehicle (or all vehicles) is of minor importance. The company is most interested in the kind of routes (difficult or easy routes) and the load of the vehicles. So, we introduced the Virtual Selection Factor (VSF) which is used in selecting a vehicle for a specific route. The VSF used a virtual route length assigned to each vehicle, instead of the actual distance.

In order to take into consideration the company's preferences we used the following objectives:

- Reduce transportation costs through better scheduling and minimization of empty kilometers.
- Fair route distribution among the available vehicles. Vehicles must be assigned: a) equal total route length throughout time (a year in our case) and b) routes of analogous difficulty.

We developed special initialization methods, mutation operators and parent replacement techniques that improved the performance of the EA.

The developed EA proved capable of finding feasible solutions that service all pickup and delivery locations, minimizing costs through better scheduling that diminish empty kilometers (kilometers covered by empty vehicles).

REFERENCES

Alba, E. & Dorronsoro, B. (2004). Solving the vehicle routing problem by using cellular genetic algorithms. In Gottlieb J, & Raidl GR (eds), *EvoCOP04: Evolutionary Computation in Combinatorial Optimization, Lecture Notes in Computer Science, 3004,* 11-20. Berlin, Germany.

De Jong, K.A. (2006). *Evolutionary Computation – A unified approach*. Cambridge, MA: Bradford Books.

Eiben, A.E. & Smith, J.E. (2008). *Introduction to evolutionary computing (Natural Computing Series)*. Heidelberg: Springer.

Gendreau, M., Potvin, J.Y. Braysy, O. Hasle, G., & Lokketangen A., (2008). Metaheuristics for the vehicle routing problem and its extensions: a categorized bibliography. In B. Golden, S. Raghavan, & E. Wasil (eds.), *The Vehicle Routing Problem – Latest Advances and New Challenges* (pp. 143-169). Heidelberg: Springer.

Jin, W-R., & Hsu, J.Y-J. (2004). A family competition genetic algorithm for the pickup and delivery problems with time window. *Bulletin of the College of Engineering, N.T.U., 90,* 88-98.

Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research, 59*(3), 345-358.

Machado, P., Tavares, J., Pereira, F.B. & Costa, E. (2002). Vehicle routing problem: Doing it the evolutionary way. In *GECCO-02: Genetic and Evolutionary Computation Conference* (pp.690-696). Morgan Kaufmann Publishers.

Mak, K.L. & Guo, Z.G. (2004). A genetic algorithm for vehicle routing problem with stochastic demand and soft time windows. *Proceedings of the IEEE Systems and Information Engineering Design Symposium* (pp. 183-190).

Ozdemir, H.T. & Mohaan, C.K. (2000). Evolving schedule graphs for the vehicle routing problem with time windows. *Proceedings of the 2000 Congress on Evolutionary Computation* (Vol. 2, pp. 888-895).