Proceedings of the 2004 Systems and Information Engineering Design Symposium
Matthew H. Jones, Stephen D. Patek, and Barbara E. Tawney eds.

A GENETIC ALGORITHM FOR VEHICLE ROUTING PROBLEMS WITH STOCHASTIC DEMAND AND SOFT TIME WINDOWS

K.L. Mak and Z.G. Guo

Department of Industrial and Manufacturing Systems Engineering
University of Hong Kong
pokfulam road
Hong Kong

ABSTRACT

This paper studies the stochastic vehicle routing problem with soft time windows (SVRPSTW). Vehicles with limited capacity are routed from the central depot to a set of geographically dispersed customers with unknown demands, predefined presence probability and time windows. The late arrival at the customer is allowed by adding a penalty to the objective value. A mathematical model is developed to describe the behavior of this kind of delivery system. A novel age based genetic scheduling algorithm is proposed as an optimization tool to solve this intractable vehicle routing problem in order to minimize the total cost. The effectiveness of the proposed scheduling algorithm is illustrated by using a set of randomly generated numerical examples. The results indicate that the proposed genetic approach is a simple but effective means for solving these problems.

1 INTRODUCTION

Designing optimal delivery or collection routes, where vehicles start and end at a single central depot, and visit a number of geographically scattered customers, is generally referred to as the Vehicle Routing Problem (VRP). Over the past two decades, VRP has become an important research topic in industrial and commercial areas. This problem can be defined by using a totally weighted complete graph G=(V,E), where V={v_1, v_2,…v_n} is a set of vertices in which v_1 is the central depot, E={(v_i, v_j): v_i, $v_j \in$ V} is a set of edges associated with a non-negative cost matrix *C* and the Euclidean distance between two vertices is usually used as that cost. It is assumed that *C* satisfies the triangle inequality, i.e. $c_{ij}+c_{jk}\geq c_{ik}$, for all v_i, v_j, v_k. Each customer is associated with a non-negative demand d_i to be collected by vehicle. A fleet of *m* identical vehicles of capacity *Q* is located in the central depot v_1. The objective of VRP is to determine a minimum cost service route for that fleet of vehicles departing from and ending at the central depot. Each customer, the vertex other than v_1, must be assigned to one vehicle and visited exactly once. The VRP is a well-known non-polynomial hard (NP-hard) problem and most techniques for solving the VRP originated from the simple Traveling Salesman Problem (TSP) (Christofides et al. 1981, Golden and Assad, 1988).

Many companies experienced difficulties in applying existing VRP concept in designing their logistics system because these mathematical models were usually formulated under unrealistic assumptions (Hicks, 1999). Most existing VRP models oversimplify the actual logistics system by assuming some system parameters (e.g. vehicle travel time, customer presence, customer demands, etc.) as deterministic values to facilitate the analysis. In most real-life distribution applications, it may not be possible to know all information about customers before designing service routes. Stochastic information occurs, and has a major impact on both how the problem is formulated and the final solution. The rise of stochastic information leads to the Stochastic Vehicle Routing Problem (SVRP), which is generally regarded as computationally intractable (Gendreau, et al. 1995).

Moreover, time window is a natural characteristic in many distribution system. Each vertex v_i may have a time window constraint [a_i, b_i] where a_i and b_i are the earliest and latest bound of the time window. Most previous studies focused on the hard time window constraint, which means the service at any customer must be taken during its time window. If the vehicle reaches to the customer v_i before the earliest bound of its time window, a waiting time w_i=(t_i-a_i) is incurred, where t_i is the arrival time. But the arrival time later than the latest bound of customer's time window is strictly forbidden. Recently, the Vehicle Routing Problem with Soft Time Windows (VRPSTW) has been studied (Koskosidis, et al. 1992, Ioannou et al. 2002) since some or all customers' time windows are not so strict but can be violated with appropriate penalties in some real-life problems. One example is bank delivery system, if a vehicle, which is sent to collect cash from different bank branches to the central bank, cannot go back to the central bank in the current day, thus losing one day's interest. But the route is feasible and the result is acceptable if the interest is not high. Usually, the time penalty is proportional to the duration of time in excess of the latest time window. But the vehicle number and/or total travel distance may be reduced. In some cases, the cost savings may be larger than the penalty cost for time violation.

The paper studies the stochastic vehicle routing problem with soft time windows (SVRPSTW), which combines characteristics of SVRP and VRPSTW. The stochastic elements in such kind of problem include stochastic demands and stochastic customers. All customer are stochastic customers and each of them has a probability, p_i, of being present. Each customer has a stochastic demand, d_i, which is governed by a known probability distribution. Readers can refer to Gendreau et al. (1996) for a comprehensive summary of the scientific literature. It is assumed that the maximum value of demand does not exceed the vehicle capacity. All demands are assumed discrete and independent. The demand at each customer is an unknown variable until the vehicle arrives at that location. If any customer is absent, its demand is regarded as zero.

The SVRPSTW is much more intricate than the classic VRP. The goal of this research is to develop an effective means to determine optimal delivery plans for SVRPSTW systems. The remainder of the paper is organized as follows. In Section 2, a two-stage recourse model for SVRPSTW is formulated to evaluate the expected routing cost for a specified solution. A new production scheduling algorithm is developed by using the techniques of genetic algorithms (Goldberg, 1989) in Section 3. In the proposed algorithm, an additional attribute, called "age," is included to enhance the computational speed and the quality of the final solution. The effectiveness of the proposed algorithm is illustrated by using numerical examples, which are described in Section 4. The conclusion follows in Section 5.

2 THE MODEL

Due to the uncertainty of demand and customer presence, a vehicle may not always follow its service route as scheduled. The SVRPSTW belongs to the class of "a priori optimization problem" which can be modeled by a stochastic programming framework and solved in two stages. The first stage solution consists of no more than m transportation routes. These routes satisfy: 1) each route starts and ends at the central depot; 2) each customer is visited exactly once by one vehicle. Whether a customer is present or not is known no later than upon leaving the preceding customer. So the next present customer in this route can be determined immediately. Since the real demand of each customer is unknown when the prior route is designed, it's probable that the vehicle capacity, Q, is exceeded during the service. In this case, route failure is said to occur. One recourse action for this exception is acted at the second stage. The vehicle returns to the depot to unload and then resumes service at the last visited customer whenever the vehicle capacity becomes exceeded; or resumes service at the next present customer along its route if the vehicle capacity is exactly attained. Moreover, the vehicle skips any absent customer at the second stage. The recourse action for route failure generates an additional cost and the skipping action achieves a potential cost saving according to the triangle inequality. The customer's presence probability and its stochastic demands have different effects on the solution. Gendreau et al. (1995) reported that stochastic customers are more difficult to handle than stochastic demands.

In some VRPs studied in some papers, an element s_i is set to the service time (loading/unloading) at customer v_i. To simplify the analysis and computation, it is assumed that the service time at any customer is zero in this paper. The generality of the problem considered is not restricted by this assumption because any SVRPSTW instance with $s_i \neq 0$ can be transformed into an equivalent instance with zero service time for all customers. Obviously, define $t_{ij}^{'}$ and [$a_i^{'}, b_i^{'}$] as the new travel time from vertex v_i to v_j and time window of vertex v_i with $s_i \neq 0$. So there exist $t_{ij}^{'} = t_{ij} + s_i$ and $\left[a_i^{'}, b_i^{'}\right] = \left[\max(a_i, a_1 + t_{1i}), \min(b_i, b_1 - t_{i1} - s_i)\right]$. The a_1 and b_1 at central depot imply the starting time of vehicles and time limitation for all routes. It is assumed that all routes have same time limitation and all vehicles start at time zero, thus a_1=0. This assumption is reasonable since it is hardly to know how many routes needed and how these routes composed in advance.

The objective is to design a first stage solution so as to minimize the expected cost of the second stage solution. The second stage cost consists of the cost of the first stage, the expected cost associated with recourse actions and the expected penalty associated with the time violation. The two-stage recourse model here is inspired by the nonlinear stochastic integer program with recourse for VRPSDC (Gendreau et al. 1995) and three-index recourse model for VRP with stochastic travel time (Laporte et al. 1992). In this paper, all distances are represented by Euclidean distance, and the velocity of all vehicles is assumed to be 1 unit. That is, numerically the travel cost c_{ij}, the travel time t_{ij} and the Euclidean distance between the vertices equal to each other.

It is assumed that there are m vehicles in the feasible solution. Let $r=\{r_1, r_2, \ldots r_m\}$ be a particular solution in which r_k is the sub tour dedicated to k^{th} vehicle. Let $r_k=\{v_1, v_i, \ldots, v_j, v_1\}$ be a particular sub tour which means the traveling sequence is $v_1 \rightarrow v_i \rightarrow \ldots \rightarrow v_j \rightarrow v_1$ serviced by vehicle k. And assuming there are only kn ($kn \leq n$) vertices in the k^{th} sub route, thus every v_{kn+1} is v_1. Relabel the vertices and the k^{th} sub route is expressed as $\{\, v_1, \ldots, v_{kn}, v_{kn+1}=v_1\}$. Let $z_k(S)$ be the expected cost of the k^{th} sub route at the second stage. So the entirely expected cost $z(S)$ is the summation of all sub routes. Define following variables used in this model:

m: the vehicle number, all vehicles are identical;

f: the cost of employing a vehicle;

c_{v_i,v_j} : the travel cost on edge (v_i, v_j);

p_{v_i} : the presence probability of vertex v_i;

$P_i^k(g)$: the expected additional cost incurred by the route failure that happened at the i^{th} customer in route k with the remaining capacity g on arrival at the i^{th} customer;

Q: the vehicle capacity;

λ : the fixed time penalty coefficient for all vertices;

h: a vector of stochastic valuables corresponding to customer demands (including zero demand). Each realization of h is called a "state of the demand";

H: the whole space of h, H is finite in this model;

$y_i^k(h)$: the lateness time when vehicle arrives at the i^{th} customer in route k in the state of demand h;

d_{v_i} : the demand value of customer $v_{i;}$

$q_{v_i,d}$: the demand probability in customer v_i whose value is equal to d;

p_h^k : the probability of the state of demand h in route k;

t_{ih}^k : the service beginning time at the i^{th} customer in the k^{th} route in the state of demand h.

The stochastic recourse model is:

$$\min z(S) = \sum_{k=1}^{m} z_k(S) + mf \tag{1}$$

The expected cost of k^{th} route is given by:

$$z_k(S) = \sum_{i=1}^{kn} \sum_{j=i+1}^{kn+1} c_{v_i,v_j} \overline{p}_{v_i,v_j} + P_2^k(Q) + \lambda E_h(\sum_{i=1}^{kn+1} y_i^k(h)) \tag{2}$$

subjected to

$$1 \le \underline{m} \le m \le \overline{m} \tag{3}$$

$$\overline{p}_{v_i,v_j} = p_{v_i} p_{v_j} \prod_{k=i+1}^{j-1} (1 - p_{v_k}) \tag{4}$$

$$P_{v_{kn}}^k(g) = 2 p_{v_{kn},v_1} c_{v_{kn},v_1} \sum_{d|(d_{vkn}>g)} q_{v_{kn},d} \tag{5}$$

$$\begin{aligned} P_{v_I}^k &= (1 - p_{v_i}) P_{v_{i+1}}^k(g) + p_{v_i} G1_{v_i}^k(g) \\ &+ q_{v_i,d|(d_{vi,d}=g)} G2_{v_i}^k(g) \end{aligned} \tag{6}$$

$$\begin{aligned} G1_{v_i}^k(g) = & \sum_{d|(d_{vi,d}>g)} q_{v_i,d} (P_{v_{i+1}}^k(Q - d_{v_i,d}) + 2c_{v_1,v_i}) \\ + & \sum_{d|(d_{vi,d}<g)} q_{v_i,d} P_{v_{i+1}}^k(g - d_{v_i,d}) \end{aligned} \tag{7}$$

$$\begin{aligned} & G2_{v_i}^k(g) = p_{v_i} P_{v_{i+1}}^k(Q) + \\ & \sum_{j=i+1}^{kn} \overline{p}_{v_i,v_j} (c_{v_1,v_i} + c_{v_1,v_j} - c_{v_i,v_j}) \end{aligned} \tag{8}$$

$$y_i^k(h) = p_h^k (t_{ih}^k - b_i)^+ \tag{9}$$

$$(t_{ih}^k - b_i)^+ = \max\{0, (t_{ih}^k - b_i)\} \tag{10}$$

In the above model, equation (1) is the objective function of the problem. Equation (2) computes the expected cost of single route. The first item of equation (2) combines of the expected first stage cost minus cost saving by skipping absent customers. Constraint (3) enforces lower and upper bounds on the number of vehicles. If $\underline{m} = \overline{m}$, vehicle number is constant. Recursive function (6) with equations (5), (7) and (8) specify the expected non-negative additional distance of the k^{th} route if the route failure occurred, given the residual capacity of the vehicle is equal to g upon arriving at customer v_i. Moreover, if the customer's demand can be delivered separately, just substitute $Q + g - d_{r_{ki},d}$ for $Q - d_{r_{ki},d}$ in equation (7). Equations (9) and (10) calculate the time penalty of the i^{th} customer along the k^{th} route in the state of demand h. This paper assumes all customers have same time requirement. Unlike the deterministic VRPTW, the arrival time maybe changes in accordance with the changing of customer demand and presence. It is worth noting that t_{ih}^k is the time when the vehicle begins to serve the i^{th} customer in route k other than reaches to that customer. If the remainder vehicle capacity is less than the customer's demand upon arriving at customer, the service cannot begin until that vehicle returns to depot to unload and goes back to that customer again. The above model reflects the true expected travel distance and the expected lateness time penalty. The penalty cost for route failure is reflected by additional travel distance. Obviously, this objective function is a nonlinear stochastic integer program with recourse and heuristic approach would appear appropriate.

3 AGE BASED GENETIC ALGORITHM

Since the SVRPSTW is a highly non-linear problem, any attempt to determine the optimal route by means of non-linear integer programming techniques is very difficult if not impossible. Genetic Algorithms (GAs), which were firstly introduced by Holland (Holland, 1975), are very easy to understand and applicable to a many kinds of problems. GA is a stochastic global search technique that imitates the natural process of evolution by using genetic operators. For many years, GA has been proved to be effective and successful in a wide variety of search domains, even for non-linear, multi-modality, high-dimensionality, discontinuity and noised mathematical models. Therefore, GA has been widely applied in solving TSPs and VRPs. In J. Grefenstette et al. (1985), GA with ordinal representation and adjacency representation were illustrated to solve TSP. The behavior of crossover was studied theoretically followed by a heuristic crossover for adjacency representation.

The GA based algorithm proposed here is named Age-GA, which is different from the canonical GA. In canonical GA, all chromosomes are replaced by their offspring after the mutation and crossover procedures as the population evolves. Due to the extremely short life span of the potential individuals, the search algorithms therefore do not have sufficient time to sample out the useful schemata from individuals. In order to ameliorate this situation, An additional attribute called age is proposed to be granted to each individual. In the Age-GA, instead of being replaced by their offspring after each iteration, individuals may grow up and generate new offspring continuously before death, and the population comprises individuals from various age-groups. The survival period of potential individuals becomes longer. Then more useful information and properties can be inherited by their offspring. The number of chromosomes generated by and survived in each age-group is determined by two sets of parameters, which are named birth rate and survival rate respectively. In order to ensure the proposed algorithm's stability, the birth rate and the survival rate of individuals must be controlled carefully, so that the population size for each are-group is keep constant throughout the search process. When the individuals gradually become older, the opportunities for them to survive and give birth decline. Their survival rates will be reduced to zero eventually, which means the individual have already completed their duties of giving birth and need to leave room for younger individuals. The following notation is used to facilitate presentation.

- L : the maximum age of an individual,
- n(*l,t*): the number of individuals contained in the population of age-group *l* in iteration *t*,
- b(*l*) : the birth rate of an individual in age-group *l*,
- p(*l*) : the survival rate of an individual in age-group *l*.

The general outline of each step of the proposed algorithm is presented below:

Step 1. Let *t* be an iteration index (*t*=0,1,2,…) with an initial value of 0.

Step 2. Generate n(*0,t*) individuals randomly and place them into the population of aged-group 0. Set n(*l,0*)=0 for all *l*=1,2,…,L.

Step 3. Determine n(*l,t+1*) by using the following equations:

$$n(0,t+1)=\sum_{l=0}^{L} n(l,t)b(l) \tag{11}$$

$$n(l,t+1)=n(l-1,t)p(l-1) \quad (l=1,2,...,L) \tag{12}$$

$$n(L+1,t+1)=0 \tag{13}$$

Step 4. Set *l*=min(L, *t+1*).

Step 5. Select n(*0,t*)b(*0*), n(*1,t*)b(*1*),…,n(*L,t*)b*(L)* individuals from the populations of age-groups *0,1,…,L* respectively, according to their selection probability. The selected individuals are the placed into a mating pool.

Step 6. Replace the population of age-group *l* by selecting n(*l,t+1*) individuals from age-group *l-1* according to their selection probabilities. If *l*>1, decrease *l* by 1 and repeat step 6.

Step 7. Pair up individuals in the mating pool and generate newborn offspring by using basic genetic operators of crossover and mutation. Replace the population of age-group 0 by those newly generated individuals.

Step 8. Check the pre-specified stopping condition. If it is satisfied, terminate the search process, and return the overall best solution as the final solution. Otherwise, increase *t* by 1, and go to step 3.

The birth rates and the survival rates must satisfy the following equation to ensure a stable genetic search process:

$$b(0)+\sum_{j=1}^{L}\prod_{l=0}^{j-1} p(l)b(j)=1 \tag{14}$$

The following Table illustrates how the birth rates and the survival rates change with individuals' ages:

Table 1. Birth Rates and Survival Rates

Age(*l*)	Birth Rate *b*(*l*)	Survival Rate *s*(*l*)
0	0.00	0.80
1	0.20	0.90
2	0.60	0.90
3	0.50	0.65
4	0.20	0.00

3.1 Chromosome Structure

The application of the proposed algorithm to solve SVRPSTW requires a string representation scheme to encode the candidate solutions in the solution space. The algorithm attempts to determine the minimum objective function value in all feasible solutions. Hence, the chromosome must contain the whole information of a route, such as the number of sub routes and the sequence of customers. The chromosome construction used here can be conveniently illustrated with a numerical example. Assuming the problem has seven vertices, from v_1 to v_7, and there are three sub routes in solution s_i. They can be represented by a list of vertices with its visited order, r_1={1,4,6,2,1}, r_2={1,7,3,1} and r_3={1,5,1}. Since all routes must start and end at the central depot, the number 1 can be omitted. For instance, the string (4,6,2) can be used to represent r_1. Similarly, r_2 and r_3 are represented by strings (7,3) and (5) respectively. The chromosome concatenates those strings and separates them by separators, which are numbers larger than *n*. The chromosome string is written as (4 6 2 **8** 7 3 **9** 5). To generalize, if there are totally *n* vertices and *m* vehicles in the problem, the number of separators is *m*-1 and they range from *n*+1 to *n*+*m*-1. The separator at the first or last position in the chromosome ought to be omitted, and successive separators ought to be regarded as one separator. The primary merits of this representation are that many classical crossover and mutation operators studied in former TSPs and VRPs papers can be applied without any modification, and the number of routes as well as the customer visiting sequence can be determined in one string.

3.2 Selection Crossover and Mutation

Selection operation is used to select the potential chromosome from the current population in accordance with their fitness values. It is the key procedure for evolvement and biases towards the fitter chromosomes and gives them higher chance of being selected. In each iteration, the number of individuals selected from age group *l* (*l*<L) to be placed into the mating pool and the survival pool is equal to n(*l,t*)b(*l*) and n(*l,t*)p(*l*), respectively. During the selection process, individuals are selected according to their respective selection probabilities. The probability of selecting a candidate solution, s_i. from the current age-group is:

$$P_s(s_i,l,t) = \frac{n(s_i,l,t)f(s_i)}{\sum_{k=0}^{P-1} n(s_k,l,t)f(s_k)} \tag{15}$$

where $f(s_i)$ is the fitness value of solution s_i. The solution space is defined as $S = \{s_{0,} s_{1,} \dots , s_{P-1}\}$. The $n(s_i,l,t)$ is the number of solution s_i contained in age-group *l* in generation *t*. The fitness function is formulated as:

$$f(s_i) = \frac{1}{z(s_i)} \tag{16}$$

where $z(s_i)$ is the objective function value of solution s_i. With reference to equation (1), the candidate solutions with lower objective function values will receive higher selection probabilities.

The crossover operation is made by exchanging a portion of genetic materials (genes) between parent chromosomes. It is a probabilistic operation which is controlled by a pre-specified parameter P_c, namely crossover probability. When it is activated, a pair of parent chromosomes is selected randomly for the mating pool, and then replaced by a new pair of offspring generated from parents. The crossover operator adopted here is the Partially Mapped Crossover (PMX). The steps of performing PMX are stated as follows:

1. Select two separated crossover points randomly. For instance, the pair of parent chromosomes are separated as P1={4 6 || **2 8 7** || 3 9 5} and P2={2 8 || **4 3 9** || 6 5 7}, where || is the crossover point.
2. Generate the offspring by swapping the numbers between two crossover points in the parents, the offspring are O1={4 6 || **4 3 9** || 3 9 5} and O2={2 8 || **2 8 7** || 6 5 7}.
3. The repeated elements before the first crossover point and after the second crossover point in the offspring are replaced by the absent elements whose original positions are between the crossover points in the parent. So the resulting offspring individuals are O1={2 6 4 3 9 8 7 5} and O2={4 3 2 8 7 6 5 9}.

Mutation operation is used to prevent the search process from converging to the local optima prematurely. It diversifies the research process to explore the candidate solution on the solution space in a random way. It is another probabilistic operation which is controlled by another pre-specified parameter P_m, namely mutation probability. The P_m is normally very small. The working steps of mutation process is best illustrated by using a numerical example. For instance, given an individual {3,5,6,7,2,4} ought to be mutated. A number between zero and the length of chromosome minus 1 is randomly generated. It is assumed 3. Then the numbers from the 3rd number, 6, to the last number, 4, in this individual are permuted. For instance if the sequence after permutation is {7,4,6,2}, the resulting individual after mutation process is {3,5,7,4,6,2}. This step guarantees that at least two numbers would be permuted or at most a whole new individual is created, which means any solution in solution space may be generated by mutation process.

The mutation process, as well as the crossover process, may change the number of sub routes in a solution and the customer sequence in a service route.

4 COMPUTATIONAL RESULTS

In this section, the genetic algorithm described in Section 3 is tested on randomly generated problem set similar with the standard problem set developed in Solomon (1987). Three different types of problems are named Rx, Cx, and RCx in which *x* designates the number of customers in the problem. The problems with the same number of customers in the same type are further denoted by an index preceded by an underline score. Such as R10_1 and R10_2 are two different problems both have ten customers in type R. In the problem set of type R, the customer coordinates are distributed by a random uniform distribution. Customers are clustered in problems of type C and semiclustered in problems of type RC. The semiclustered problem is the one that contains a mix of randomly generated and clustered data.

It is assumed that there are no more than 3 homogeneous vehicles located in the central depot in all problems. The cost of employing a vehicle is set to be 1 and the capacity, *Q*, is assigned to 50. All graphical coordinates in test problems are generated within a $[0, 100]^2$ square. And the traveling cost matrix is assumed as symmetric so that $c_{ij}=c_{ji}$. The presence probability p_i of each customer v_i (i>1) was generated in interval (0 1) independently. Each customer may have 2, 3 or 4 dissimilar demand quantities and each demand quantity is generated in the range [1, 20] randomly. The probability of each demand level is generated in (0, 1) according to random distribution. Of course, the summation of probabilities of all demand levels in any customer is definitely equal to 1.

So far, the geographical and demand data were given. The time windows for all vertices, including central depot, are generated at the last step. These time windows were designed in such a way that at least a few routes without route failure are feasible even in hard time constraints. Firstly, randomly design a solution containing *m*-1 service tours, and make sure every vehicle returns to the central depot before the upper bound of the time window of the central depot. Secondly, the time window constraints were generated by letting the arrival time at the customer be the center of its time window. To create the width of the time window for customer v_i, a random integer number is generated as half of the width of the time window. At last, these time windows are manually adjusted to ensure that $b_i \ge a_1 + c_{1,i}$, $a_i \le b_1 - c_{1,i}$, $b_i > a_i$ where a_i and b_i are the integers for vertex v_i.

After generating the example data, the parameters of the Age-GA need to be determined. The crossover and mutation probabilities are set to be 0.8 and 0.05 respectively. The initial population size is set to be 60. The stop condition is 100 iterations. The maximum age of any individual is fixed at 4 and the birth and survival rates were presented in Section 3.

For the sake of comparison, the test problems were also solved by using a Canonical GA (CGA), which used ordinal representation, single-point crossover and uniform order-based mutation. The population of CGA is set to 30, other parameters are same as the corresponding parameters in Age-GA. A Pentium III-866MHz with RAM 512M based computer is used to implement both algorithms.

Table 2. Results of Age-GA

Instances		Age-GA		
Names	λ	Total cost	Lateness	Time(s)
R10_1	1	276.12	13.06	6
	10	322.06	0.82	2
	100	309.63	0.13	4
R10_2	1	344.41	4.21	15
	10	387.55	0.086	5
	100	413.93	0.11	9
C10_1	1	336.46	5.82	2
	10	351.52	1.71	3
	100	465.33	1.23	3
C10_2	1	316.71	29.56	11
	10	375.51	1.79	6
	100	489.24	1.4	5
RC10_1	1	354.86	33.32	5
	10	372.13	2.54	2
	100	513.19	1.59	3
RC10_2	1	368.36	8.5	8
	10	434.42	4.07	6
	100	688.53	2.83	7
AVERAGE		395.55	6.26	5.67

Table 3. Results of CGA

Instances		CGA		
Names	λ	Total cost	Lateness	Time(s)
R10_1	1	294.94	2.64	3
	10	341.04	3.18	5
	100	429.01	0.86	3
R10_2	1	358.79	15.42	12
	10	398.78	2.74	10
	100	513.19	1.2	5
C10_1	1	339.07	14.63	3
	10	355.19	7.47	2
	100	523.68	1.46	2
C10_2	1	333.89	11.57	10
	10	390.95	5.9	6
	100	882.47	4.32	5
RC10_1	1	355.36	6.63	5
	10	376.42	1.04	3
	100	790.11	4.13	5
RC10_2	1	374.18	6.67	6
	10	466.23	1.02	6
	100	677.48	2.89	5
AVERAGE		455.60	5.21	5.33

Eighteen problem instances were generated in this paper. These test problems highlight the time penalty coefficient that impacts the behavior of soft time windows constraint. Table 2 presents the final results provided by Age-GA and Table 3 presents the final results solved by CGA. The final result is the overall best solution in search process. The report rules contain: the expected total cost of the final solution (Total cost), the computational time consumed (Time) and the expected total lateness time in the final solution (Lateness), which is calculated by dividing total expected time penalty by λ.

Upon analyzing the data in result Tables, it can be deduced easily that the Age-GA clearly outperforms the CGA about 15%. Nearly all results obtained by Age-GA are better than the corresponding results obtained by CGA. The computational times consumed by two algorithms are similar, although the population size of Age-GA is larger than the population size of CGA. It is because the crossover and mutation operations are only executed in age-group 0 in Age-GA so that it need not to recalculated the objective values of individuals in other age-groups.

It is worth noting that the increase of time penalty coefficient, λ, usually leads to the increase of expected total cost, and decreases the expected total lateness time of final result. When λ is small, the expected travel distance is a significant part in total objective value. The primary goal is to reduce the travel distance so that commonly more customers have to be planned for one vehicle. Although customer time windows have been violated from time to time in this circumstance, the expected total time penalty is still comparatively small. When λ rise, the expected time penalty become more and more important in total objective value. The primary goal is changed to avoid the time window violation as much as possible. Typically more vehicles are used and fewer customers are arranged in one service route, which increases the expected total travel distance. In this case, each customer can be served earlier averagely, which leads to decrease the expected total lateness time. It is believed if λ is large enough, the final solution may satisfies all customers' time windows if such solution exists. The problem has been changed from SVRP with soft time windows to SVRP with hard time windows eventually.

Figure 1 shows the convergence behavior of the Age-GA for a typical search process of the problem RC10_2 with $\lambda = 10$. The search process converges very rapidly in the first few generations then slows down the convergence speed in subsequent generations. It is because the initial population is generated totally randomly, the objective function values are often high. And most regions of the solution space are unexplored in the beginning. Hence, it is relatively easy to discover new promising regions and produce good individuals. With removing the poor performed individuals, the mean objective value of whole population also drops fast. As the search process going on, more promising regions have been explored and it is harder to find the better solutions. Therefore, the speed of convergence decreases. When the Age-GA attempts to exploit promising regions, the mutation operation plays a very important role at this stage to prevent the search process from being trapped into local optima. But the mutation operation may also destroy the good individuals and produce bad individuals. This causes the wave of mean objective function value.

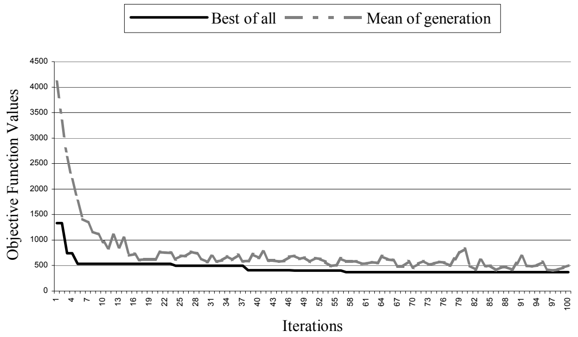

Figure 1. Typical convergence process of the Age-GA

A sensitivity analysis on the genetic parameters has also been studied in this paper. It indicates that the premature convergence may be avoided and better final solution may be obtained by increasing the population size. However, this could severely lengthen the computational time, especially for such difficult type of problem. A large mutation probability can reduce the convergence speed of the genetic search process and avoid premature convergence. But the mutation probability can not be too high, otherwise, the genetic search process will behave like a random search process. In addition, increasing the crossover probability will not improve the quality of final solution, neither affects the convergence speed. It should not be too larger nor too small according to experiences.

5 CONCLUSION

This paper has presented the use of an age based genetic algorithm as a general methodology to solve stochastic vehicle routing problem with soft time windows. A nonlinear stochastic integer program has been formulated to provide the objective function for optimization algorithms. A numerical chromosome presentation is chosen to reveal both the number of routes in solution and the customer sequence in each route. Eighteen randomly generated problems have been used to evaluate the effectiveness of the proposed algorithm. A canonical genetic algorithm has also been applied to solve the same group of numerical problems. The results have shown that the optimal or near-optimal solutions can be obtained by using Age-GA with reasonable computational time. Comparison of the results obtained by using Age-GA and CGA shows that Age-GA is superior to CGA. The success of Age-GA seems mainly due to its age property that allows good individuals to survive longer and gives them more chance to generate offspring. Moreover, for the same amount of computational effort, it is possible to use a larger population size in Age-GA than in CGA.

This paper also analyses the behavior of SVRPSTW and the impacts of time penalty coefficient on final solutions. The results show that some elementary properties of deterministic VRPTW are no longer kept in the stochastic case. In addition, sensitivity analysis has been performed to study the effects of the various genetic parameters on the convergence behavior of the proposed algorithm. The results reveal that the population size and the mutation probability are two key factors that govern the convergence rate and the quality of final solution.

REFERENCES

Christofides, N., A. Mingozzi, and P. Toth. 1981. Exact Algorithms for the Vehicle Routing Problem, Based on Spanning Tree and Shortest Path Relaxations. *Mathematical Programming* 20: 255-282.

Gendreau, M., G. Laporte, and R. Séguin. 1995. An Exact Algorithm for the Vehicle Routing Problem with Stochastic Demands and Customers. *Transportation Science* 29(2): 143-154.

Gendreau, M., G. Laporte, and R. Séguin. 1996. Stochastic Vehicle Routing. *European Journal of Operational Research* 88: 3-12.

Goldberg, D.E. 1989. *Genetic Algorithms in Search, optimization and Machine Learning,* New York: Addison-Wesley.

Golden, B.L. and A.A. Assad, 1988. *Vehicle Routing: Methods and Studies*. Amsterdam: North-Holland.

Grefenstette, J. R. Gopal, B. Rosmaita and D.V. Gucht, 1985. Genetic Algorithm for the Traveling Salesman Problem. *Proceedings of the 1st International Conference on Genetic Algorithms and Their Applications*: 160-165.

Hicks, D.A. 1999. The state of supply chain strategy. *IIE Solutions* 31(8): 24-29.

Holland J.H. 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan press.

Ioannou, G. M. Kritikos, and G. Prastacos. 2003. A problem generator-solver heuristic for vehicle routing with soft time windows. *The International Journal of Management Science* 31: 41-53.

Koskosidis, Y.A. W.B. Powell, and M.M. Solomon. 1992. An Optimization-Based Heuristic for Vehicle Routing and Scheduling with Soft Time Window Constraints. *Transportation Science* 26(2): 69-85.

Laporte G., F. Louveaux and H. Mercure. 1992. The Vehicle Routing Problem with Stochastic Travel Times. *Transportation Science* 26(3): 161-170.

Solomon, M.M. 1987. Algorithms for the vehicle routing and scheduling problems with the window constraints", *Operations Research* 35(2): 254-265.