

Proposal and Evaluation of “Selfish-Gene with Limited Allowance” Type GA for Solving Constraint TSP

Takashi Onoyama*, Sen Kubota*, Yoshio Taniguchi** and Setsuo Tsuruta***

* Hitachi Software Engineering Co., Ltd.

6-81, Onoe-cho, Naka-ku, Yokohama, 231-0015, Japan

**Hitachi Microsoftware Systems, Inc.

Yoshida-cho Totsuka Yokohama, 244-0817, Japan

Setsuo Tsuruta*, Takashi Onoyama** and Yoshio Taniguchi***

*** Systems Development Laboratory, Hitachi, Ltd.

1099 Ohzenji Asao-ku Kawasaki-shi, 215-0013, Japan

e-mail: {onoyama, kubosen}@ori.hitachi-sk.co.jp, tanigu@hitachi-ms.co.jp, tsuruta@sdl.hitachi.co.jp

Abstract

For realizing large-scale distribution network simulation applicable to supply-chain management, it is required to solve hundreds of time constraint large-scale (100 cities) Traveling Salesman Problem (TSP) within interactive response time, with practicable optimality. Thus, a "selfish-gene with limited allowance" type GA is proposed. Here, each gene of an individual satisfies only its constraints selfishly, disregarding the constraints of other genes in the same individual. Further, to some extent, even individuals that violate constraints can survive over generations and are given the chance of improvement. Our experiment proves that this method gives expert-level solutions for time constraint large-scale TSPs within several seconds.

1. Introduction

The simulation for large-scale efficient distribution networks is an important matter not only from the economical viewpoint but also from the social aspects such as the curtailment of CO₂ and auto exhausts. Recently, in addition to distribution efficiency itself, transportation strongly specifying time-constraints, such as “just in time”, are increasingly required in order to improve productivity through supply chain management. Though human judgement is indispensable in considering the above social aspects, even experts cannot easily generate optimal routes under severe time-constraints. Error rates are sometimes as high as 50% when compared to optimal solutions. This causes many problems.

Thus, a computer needs to create more optimal routes within interactive response time (e.g. less than 10 seconds), and human experts have to evaluate the social aspects. Since a large-scale distribution network has hundreds of distribution routes, one route has to be created within a few seconds. And as the creation of each route is equivalent to a TSP (Traveling Salesman Problem) of tens of cities, an approximate solving method that enables both interactive responsiveness and optimality is required.

As a method for solving distribution problems with delivery time constraints, a method based on IP (Integer

Programming) (Ikegami & Niwa 1995) has been proposed. Although this method can solve these problems optimally, it cannot satisfy the above-mentioned responsiveness. Solving methods utilizing tabu search method or Simulated Annealing (SA) (Yamamoto & Kubo 1997) (Hooker & Nastroj 1995) also have problems of long calculation time.

Our proposed method aims at solving TSPs with strict time constraints, guaranteeing the responsiveness and practical optimality with error rate below 20%. In the next section, a time-constraint distribution scheduling and its technical problems are described. In section 3, a solving method that meets the above problem is proposed. In section 4, experimental results are indicated as to the proposed method. In section 5, the effectiveness of this method is discussed.

2. Field Example of TSPs with Constraints and its Technical Problems

2.1 Large-scale Distribution Network Simulation

A large-scale distribution network is shown in fig. 1. Parts for production are delivered from parts manufactures directly to factories or through deposits. However, these parts are not delivered to a factory or a deposit directly by each of parts manufacturers. Instead, a truck collects these parts from the manufacturers. This improves distribution efficiency, contributing to the curtailment of distribution expenses and to the reduction of the volume of auto exhausts.

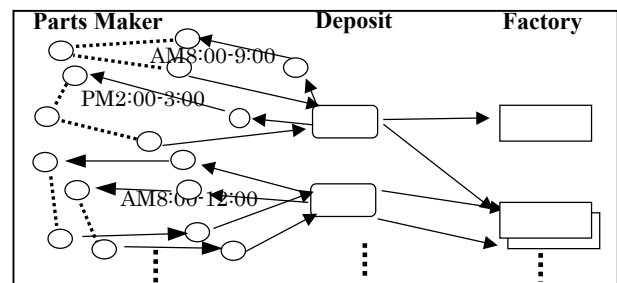


Figure 1. Large-scale Distribution Network

- | |
|---|
| <ol style="list-style-type: none"> 1) Set Conditions <ol style="list-style-type: none"> i) Setting Factory Locations ii) Setting Depository Locations iii) Setting Parts Maker (Supplier) Locations 2) Create delivery routes and calculate costs 3) Decide Optimal Distribution Network |
|---|

Figure 2. Simulation Process

Since there are about one thousand parts manufacturers, hundreds of trucks are needed for this operation, and each of them visits tens of (max. 100) parts manufacturers. About 20% of the delivery points (mainly parts manufacturers) have the specified delivery time range (namely time constraints) as shown in fig.1. In some cases, strict or strong constraints (e.g. within certain one hour) are specified, while weak constraints (e.g. in the morning or in the afternoon) are specified in other cases.

In constructing above-mentioned efficient large-scale distribution network, it is necessary to grasp the total cost of distribution under various conditions by repeating the simulation process as shown in fig. 2. However, there are hundreds of parts manufacturers, dozens of deposits and more than ten factories. Therefore, there are hundreds of distributing routes, each of which tours scores of parts manufacturers (max. 100), starting from one of the deposits or factories. Thus, with each simulation, delivery route creation is repeated several hundreds of times for conditions manually set up, and a person in charge totally decides the network's optimality.

2.2 Technical Problems

The above delivery route generation problems are theoretically equivalent to TSPs with time constraints. Thus, to optimize such a large-scale distribution network, the following technical problems have to be solved.

1) Optimality with time constraint satisfied: In case time constraints are strict, even the solutions of human experts have sometimes around 50% of deviation from the optimal solutions. To overcome intensive competitions in distribution services, the guarantee below 20% of maximum errors and below 10% of average errors is required.

2) Responsiveness: Since truck route generation requires the consideration of not only cost but also social factors, it is indispensable for human experts to manually check truck routes generated by machines. For such interactive use of the system by human experts, response time of at least within 10 minutes is indispensable. Since there are usually around hundreds of trucks in a large-scale distribution network, one truck route has to be generated within a few seconds.

3. Proposed Solving Method

3.1 The Basic Idea of Proposed Solving Method

In order to guarantee both responsiveness and optimality in solving TSPs with time constraints as described in section 2, a GA method, which we call "selfish gene with limited allowance type GA", is proposed. Basically this method consists of the following two ideas:

The first idea is a new concept of "selfish-gene", which is different from Dawkins' "selfish gene". The original concept of "selfish gene" by R. Dawkins (Dawkins 1989) lies in the following idea. Namely, all genes of an individual cooperate with each other, aiming at the prosperity of the individual for its selfishness. Strictly speaking, all genes constituting a chromosome cooperate with each other, aiming at the prosperity or reproduction of the chromosome to satisfy the chromosome's selfishness. We further developed this idea in that each gene on a chromosome or an individual acts selfishly to optimize only itself or the relationship between itself and its surroundings locally, without considering the totality of the individual or other genes of the same individual. When the constraint is quite severe and/or higher optimality is required, it is natural to imagine that such a local selfish action occurs.

The second idea is a concept of limited allowance. Namely, to some degree, "God" or "natural system" admits reproduction or prosperity of a chromosome or an individual, each constituent of which is a selfish gene. However, at the end, through reflection or punishment at some appropriate generations, only individuals, which satisfy the objectives or constraints, can survive over generations.

The above two ideas were obtained to find an efficient GA method that can optimally solve strong or strict time constraint TSPs. Concretely speaking, in the GA method for solving time-constraint TSPs, each gene satisfies only the time constraints directly related to itself, and it doesn't consider satisfying the time constraints of other genes in the same chromosome or individual. Further, each such individual or chromosome, each constituent of which is a selfish gene, is allowed to survive over generations under the following conditions. That is to say, the chance of globally satisfying constraints (we call the chance of reflection) is given through changing the structure (e.g. the insertion order in NI method) of each constituting gene per GA's operation such as crossover. Furthermore, an individual (strictly speaking, a chromosome or a unit genes' group), is not killed until some particular generations even if it includes a selfish-gene that cannot globally satisfy constraints.

In this way, through allowing the selfishness of each constituting gene, and through not examining the global constraints' satisfaction except for some particular generations, GA's operation time is shortened. And, through allowing various types of individuals to survive, more optimal solutions are found. Thus, both high optimality and speedy responsiveness are obtained.

In the real world of living things, a unit genes' group or an individual that perfectly adapts to the environment does not suddenly emerge in their evolving stages. The primitive and imperfect living things, which surely violate some rules or constraints, evolve slowly to adapt to their environment. Likewise this phenomenon exists in the optimization of GAs. If we try to continue satisfying all constraints at generations, the population size often shrinks or the system falls into local minimum. Efficient optimization cannot be attained in this fashion. When constraints are very severe, the whole population can be destroyed in the worst case. This is to say, if a population consists of only individuals that satisfy constraints, a remarkable improvement is not

expected. Yet, there is a risk of falling into local minimum. For such reasons, we also give a limited chance of survival, reflection or reparation to the individuals violating constraints.

3.2 Architecture of the GA

Now, the GA architecture is proposed, which realizes the above concept to solve TSPs with time constraints more optimally and speedy.

(1) Insertion of selfish nodes:

As an approximate solving method for TSPs, the authors realized a highly optimal and high speed solving method of GA (Onoyama & Tsuruta 2001), utilizing NI (Nearest Insertion) method embedded with knowledge in crossover and mutation operations. To realize the above concept through improving this method, an insertion method called "selfish NI method" is applied to put nodes into a tour. In this method, a node is inserted into a tour, using NI method, if the node satisfies only its constraints. That is, the influence on the other nodes is not considered.

(2) Crossover

Here, through applying "selfish NI method" described in (1), knowledge incorporated NI crossover operation already described (Onoyama & Tsuruta 2001) is extended to efficiently deal with time constraints. This extended crossover operation consists of the following steps: 1) Determine randomly a crossover point in one of the parents' chromosomes. 2) Remove nodes in a partial tour corresponding to genes in latter half of the crossover point of the chromosome. 3) Into the remaining partial tour, insert the removed node having no time constraint in the same order as another parent chromosome through NI method. 4) Following the same procedure as 3), insert removed nodes having time constraints through selfish NI method.

(3) Mutation

The mutation operation of GA is useful to escape from local minima and obtain the optimal solution. However, simple mutation operation using only random process without local search method on problem-oriented knowledge is not efficient for GA's convergence to the solution. Thus, we developed the block type mutation (Onoyama & Tsuruta 2001).

This operation is extended also through applying selfish NI method described in (1), as follows. 1) Select a mutation node (gene) randomly out of a tour (chromosome), and remove the node together with its neighboring nodes. The size of the neighborhood is determined, based on the problem-oriented knowledge. 2) Then, through NI method, insert the removed nodes having no time constraints into the remaining nodes (genes) of the tour (chromosome). 3) Lastly, through the selfish NI method, remove nodes having time constraint through selfish NI method.

(4) Selection

The selfish NI applied crossover and mutation, such as described in (2) and (3), may leave some individuals that violate time constraints. Therefore, the following two selection methods are proposed in order to handle such violations.

(a) Check and repair per generation

On each generation of GA, check whether all individuals satisfy the time constraint. Then, for each of individuals whose time constraints are violated, remove from the tour the nodes (genes) that violate time constraints, from the tour and repair the tour through reinserting the nodes by means of selfish NI method. In case time constraints cannot be satisfied even through this repair procedure, the individual is not selected for the next generation.

(b) Check on only specified generations

If time-constraints are checked and violations (or individuals) are repaired per generation, the potential of evolution can be narrowed down and the possibility of falling into local minimum is increased. Worse, frequent time-constraint checks and reparations require much time, which is disadvantageous to the responsiveness. Thus, the above checks and reparations are restricted only to specified generations. However, individuals that do not satisfy time constraints can be improved through the next generation's crossover and mutation using selfish NI method. Therefore, better optimization can be realized in this method with less checking overhead.

3.3 Proposed Solving Method

The following two methods are proposed, based on the architecture discussed in 3.2.

(1) Narrow-minded (punishing per generation) selfish-gene type GA

For this type, selfish NI method is applied to crossover method and block type mutation. The time constraints check and reparation are done per generation. Furthermore, just after the selection in every generation, the population consists of only individuals that satisfy time constraints.

(2) Selfish-gene with limited allowance type GA

For this type also, selfish NI method is applied to crossover and block type mutation. The time constraints check is carried out only on some specified generations, or at least on the last generation. Thus, this solving method allows even individuals violating time constraints to reproduce themselves freely.

In the next section, experiments to evaluate the effect of the above two selfish-gene types of GA are performed. These tests compare them with the conventional non-selfish (gene) type GA. Non-selfish type GA applies conventional non-selfish NI method to crossover or mutation operation.

This NI method inserts a node into the position that satisfies not only its constraints but also the constraints of every other node in the same individual. In case such positions of nodes are not searched out, crossover or mutation operations fail and the children are killed as fatal genes.

4. Evaluation Experiment and its Effect

4.1 Experiment

A PC equipped with an Intel Pentium III (600 MHz) processor and 512 MB memory is used for the experiment. Considering the scale of distribution network in question, 100 cities TSPs, each of which includes 20 cities having different types of delivery time constraints, are prepared. Namely, there are two types of time constraints: one is a

half-day time window type that is not so severe (weak time constraint), and the other is an one-hour time window type that is severe (strong time constraint). Thus, 50 TSPs with strong time constraints and another 50 TSPs with weak time constraints are randomly generated. These 100 TSPs are solved 100 times, each with the three types of solving methods described in 3.3 to check the responsiveness as well as the optimality.

When generating the initial population, nodes are inserted into a tour, following the random order by means of selfish NI method. Then, for each of the individuals among the initial population, its nodes (genes) violating time constraints are removed from the tour and reinserted into the tour by means of selfish NI method to repair. Concerning the selfish-gene with limited allowance type GA and narrow-minded selfish-gene type GA (both of which belong to selfish-gene type GA) the individuals that could not resolve the violation of time constraints through reparation, are included in the initial population to reach a specified population size. In the selection operation, highly evaluated individuals are selected out of the population including both parents' and children's. Thus, even if all children are fatal genes, their parents' genes survive for next generations. Further, to enhance the evolution efficiency, only one individual is selected, when the same individuals are generated.

In this experiment, the number of generations is 50, and the population size is 100. Each of both crossover rate and mutation rate is 20 % respectively.

4.2 The Results of the Experiment

Table 1. Response time of each method

# GA Method	Weak constraint	Strong Constraint
1. Narrow-minded selfish-gene type	3.9 sec (Ave) 5.8 sec (Max)	3.1 sec (Ave) 17.1 sec (Max)
2. Selfish-gene with limited allowance	2.5 sec (Ave) 3.0 sec (Max)	2.6 sec (Ave) 6.4 sec (Max)
3. Non-selfish type	3.7 sec (Ave) 5.7 sec (Max)	4.8 sec (Ave) 25.7 sec (Max)

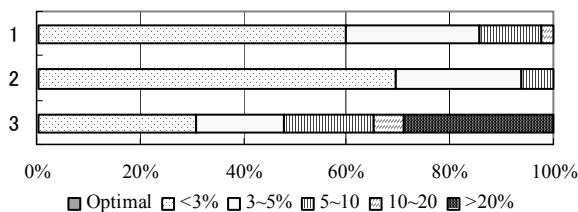


Figure 3. Optimality under weak time constraints

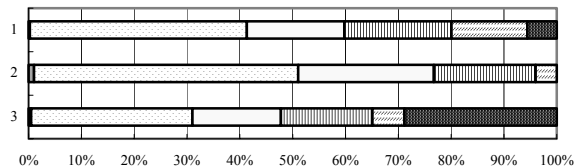


Figure 4. Optimality under strong time constraints

Tab.1 shows the response time of each method. Fig3 indicates the optimality under weak time constraint such as a half day time window. Fig. 4 shows the optimality under the strong time constraint such as a one-hour time window.

5. Evaluation

5.1 Evaluation based on Experimental Results

According to the experimental results, the response time of each method is all between 2.5 and 6 seconds under the weak time constraint. However, under the strong time constraint, maximum response time exceeded 17 seconds for narrow-minded (punishment per generation) selfish-gene type GA, though response time was still at most 6.4 seconds for selfish-gene with limited allowance type GA. Furthermore, with the conventional non-selfish type GA, the maximum response time exceeded 25 seconds. Thus, the latter two methods are unable to realize the interactive responsiveness, as needed in our distribution simulations.

Concerning the optimality, selfish gene with limited allowance type GA had the distinguished results irrespective of the severity degree of the constraints. In case of the weak time constraint, the error rates of this method were within 5 %.

On the other hand, as for both narrow-minded selfish-gene type GA and conventional non-selfish type GA, maximum error rates exceeded 10%. As far as the strong time constraint is concerned, the highest error rate of the selfish gene with limited allowance type GA was under 20 % and its error rate under 10% was 97% of the cases in the experiments. However, as to narrow-minded selfish-gene type GA, the number of cases with error rates over 20 % exceeded 5 % of the whole cases of the experiment. And as to conventional non-selfish type GA, the case whose error rate is over 20 % exceeded 30 % of the whole cases of the experiments.

(1) The effect of selfish-gene type GA

Conventional non-selfish gene type GA checks all nodes having time constraints, every time a node is inserted into a tour and this type GA takes much time to obtain optimal solutions. This method's check for time-constraint satisfaction requires the calculation step proportional to the following expression per generation.

$$((\text{group size}) * (\text{mutation rate} + \text{crossover rate}) * (\text{individual's length}))$$

As for solution optimality, the result was not so bad in case of weak constraints, but in case of strong or severe constraints (narrow time window), the case whose maximum error rate is over 20% exceeded 30% of the tested cases. This is because it is very difficult for all genes of individuals to satisfy constraints in case the constraints are strong. As a result, too many individuals in the population die. For example, about 95 % of all individuals in a population died at the end of the 5th generation. For such a reason, local minimum prevails and necessary optimization level is not attainable.

(2) The effect of the limited allowance

The response time of narrow-minded selfish-gene type GA

was twice as much as that of selfish-gene with limited allowance type GA.

As for optimality, both methods were about the same level in case of weak constraints. However, in case of strong constraints, narrow-minded selfish-gene type GA's error rate occasionally exceeded 20%, while selfish-gene with limited allowance type GA's maximum error rate was under 20%. As to narrow-minded selfish-gene type GA, most of the individuals (unit genes or chromosomes) created at the initial solution stage, crossover, or mutation, were killed, when the time constraints were strong. Only below 10 % of those individuals remained alive at the end of the 10th generation. In this way, narrow-minded selfish-gene type GA's population size narrows down to an extent that causes local minimum. Thus, it fails in realizing the desirable optimization.

In contrast with this, selfish-gene with limited allowance type GA's error rate was below 10 %, which was the best score among the three methods experimented here. This solving method lets individuals, which cannot satisfy constraint, survive over generations. Further, in this method, the chance for reparation from constraint violation is given to individuals through selfish NI used in crossover and mutation for future generations. It is considered that this is the reason why this method could efficiently obtain highly optimal solutions. According to the above consideration 1) and 2), in case of optimization problems having strong constraints, the appropriate evaluation criteria in GA's selection are important. For example, even though it is not possible or not done perfectly to achieve one's reflection or reparation within one generation, there is always room for it in the succeeding generations. And due to selfishness of each gene to continuously try to satisfy its constraints, a better result can be often obtained after some generations. There seems a similar law in the natural world's selection (natural law or God's wisdom).

5.2. Comparison with other Methods

Several methods to solve time-constraint TSPs are proposed (Ikegami & Niwa 1995) (Hooker & Natraj 1997)(Tan et al. 2001).

The solving method integrating IP with heuristics (Ikegami & Niwa 1995) marked high optimality of below 3% error for 50 cities' TSPs. However, since this method needs 30 seconds for such scale TSPs, it cannot be applicable to the case that requires interactive response time.

Theoretically, SA (Yamamoto & Kubo 1997) is said to be able to search almost optimal solutions by decreasing the risk of falling into local minimum. But practically, in various problem patterns, it is very difficult to adjust SA's parameters such as cooling speed. Furthermore, SA usually takes a long calculation time to get above-mentioned theoretical near-optimal solutions. Tabu Search (Hooker & Natraj 1995) usually needs also a long calculation time to get practically optimal solutions. In some of our experiments, it took about 400 milliseconds for SA to solve a TSP of 15 cities and about 40 seconds to solve a TSP of 100 cities. Therefore, these methods are not suitable for repetitive simulations including interactive human judgements as needed in our applications.

6. Conclusion

As a solving method of TSPs having constraints, selfish-gene with limited allowance type GA was proposed and proved to give responsiveness and optimality necessary for a large-scale distribution network's interactive simulation. This solving method based on the following two ideas.

1) Each gene, which constitutes "unit genes' group" (an individual), does not care about constraints of all other genes within the individual. Thus, it occasionally tries so much to optimize only itself or the relationship with its neighboring genes locally that it acts selfishly. This is our new idea of "selfish gene" which is different from R. Dawkins' "selfish gene", since he does not explicitly refer to the selfishness of each gene constituting a chromosome (Dawkins 1989).

2) The second idea is that such a unit genes' group generated through the above selfish way is allowed to live over generations to some extent. And, because of reflection or reparation and strict punishments in some generations or at least in the final generation, the unit genes' group (namely chromosome or individual) can gradually and finally satisfy its constraint.

The idea of "selfish-gene" enables to speed up the calculation time so that the response time can be guaranteed. Another idea of "limited allowance" enables to avoid falling into local minimum through selecting individuals violating constraints and giving them chances of reflection or reparation over generations.

Our experimental results showed that the proposed method makes it possible to obtain expert-level solutions within interactive response time (a few seconds) for large scale (eg.100 cities) TSPs having strong time constraints. Thus, this method is considered to be applied to interactive simulations aiming at the construction of large-scale optimal distribution networks.

References

- Ikegami, A.; and Niwa, A. 1995. The Vehicle Routing Problem with Time Constraints, *Journal of Operations Research of Japan*, Vol.38, No.1, pp.107-123.
- Onoyama, T.; Kubota, S.; Oyanagi, K.; and Tsuruta, S. 2001. Knowledge-embedded Multi-stage Genetic Algorithm for Interactively Optimizing a Large-scale Distribution Network, *FLAIRS'01*, pp.166-170.
- Yamamoto, Y.; and Kubo, M. 1997. *Invitation to Traveling Salesman Problem*, Asakura Syoten, Tokyo.
- Hooker, J.H.; and Natraj, N.R. 1995. Solving a General Routing and Scheduling Problem by Chain Decomposition and Tabu Search, *Transportation Science*, Vol.29, No.1, pp.30-44.
- Tan, K.C.; Lee, T.H.; Ou, K.; and Lee, L.H. 2001. A Messy Genetic Algorithm for the Vehicle Routing Problem with Time Window Constraints, *Proc. CEC2001*, pp.679-686.
- Dawkins, R., 1989, *The Selfish Gene*, Oxford Univ. press.