Selfish Constraint Satisfaction Genetic Algorithm for Planning a Long-distance Transportation Network

Takashi Onoyama and Takuya Maekawa
Hitacih Software Engineering Co., Ltd., Tokyo, Japan
E-Mail: {onoyama, taku}@hitachisoft.jp

Yoshitaka Sakurai and Setsuo Tsuruta
Department of Information Environment, Tokyo Denki University, Inzai, Japan
E-Mail: {ysakurai, tsuruta}@sie.dendai.ac.jp

Norihisa Komoda
Graduate School of Information Science and Technology, Osaka University, Suita, Japan
E-Mail: komoda@ist.osaka-u.ac.jp

***Abstract*—To build a cooperative logistics network covering multiple enterprises, a planning method that can build a long-distance transportation network is required. Many strict constraints are imposed on this type of problem. To solve this problem efficiently, a selfish-constraint-satisfaction genetic algorithm (GA) is proposed. In this type GA, each gene of an individual satisfies only its constraints selfishly, disregarding constraints of other genes even in the same individual. Further, to some extent, even individual that violates constraints can survive over several generations and has the chance of reparation. Moreover, a constraint pre-checking and dynamic penalty control methods are also applied to improve convergence of GA. Our experimental result shows that the proposed method can obtain an accurate solution in a practical response time.**

Index Term — genetic algorithm, logistics network, cooperative logistics, vehicle routing problem

I. INTRODUCTION

Logistics cooperation covering multiple enterprises is an effective method for improving logistics efficiency of parts procurement [1]. However, this cooperation forms a "cooperative logistics network" consisting of several mutual sub-networks such as "parts-collection networks" covering parts suppliers and depots (distribution centers) and a "long-distance transportation network" covering depots and factories. Thus, construction of this type of cooperative logistics network is therefore a very complicated task. Accordingly, usually this task is divided into several phases [2, 3].

We already proposed a two-phased ("construction phase" and "scheduling phase") planning method for optimizing a cooperative logistics network [4]. In the construction phase, which corresponds to strategic or tactical level planning, we make a parts-procurement plan corresponding to a production plan and assign resources such as drivers and vehicles to sub-networks [5, 6]. Next, in the scheduling phase, an actual transportation schedule to evaluate the operability of the network and to evaluate accurate logistics cost based on the number of used vehicles is made.

A simple mathematical optimal plan is usually not readily accepted because of conflicts that arise in the different enterprises. A human expert must thus evaluate the plan from many aspects and coordinate it with consideration of many non-systemized conditions. To build logistics plans for the scheduling phase, solution methods for scheduling a long-distance transportation network and for depot-centered parts-collection networks which provide at least human-expert-level accuracy within interactive response time are required. We already proposed a VRP (Vehicle Routing Problem) [7, 8] solution method for parts-collection networks [9]. In this paper, we propose a scheduling method for a long-distance transportation in the cooperative logistics network.

A long-distance transportation network is not only directly connected to the production schedules of factories but also connected to many parts-collection sub-networks. Moreover, this network contains many transportation points (depots and factories) dispersed all over the nation. And on the network, a vehicle distributes and collects parts at the same time. Stricter constraints are thus imposed on this type of scheduling problem than those on a normal VRP.

As a method for solving VRP, IP (Integer Programming)

has been applied [10]. Although, this method can obtain the optimal solution, it cannot satisfy the practical response performance.

The sweep method [11] is a high-speed solution adopting comparatively simple heuristics. However, it is difficult to solve a tough-constraint problem with practical accuracy.

A mathematical model such as a minimum-cost-flow model with a time period is also applied for constructing a large-scale transportation network [12, 13, 14]. This method is suitable for determining the transportation capacity of each sub-network and for determining the allocation of the cost to each enterprise. However, it is impossible to evaluate the accurate logistics cost based on the number of used vehicles.

Meta-heuristics such as the tabu search or SA (Simulated Annealing) [15] combined with a local search method such as cross-opt [16, 17] have also been proposed, and these methods can obtain highly accurate solutions [18, 19]. However, it is difficult to apply these methods to our problem because a vehicle picks up and distributes parts at the same time. Moreover, some parts are directly transported between factories, that is, not through depots. Thus, we cannot apply these local search methods to our problem.

A method of applying GA (Genetic Algorithm) that introduces route-exchange genetic operation has also been proposed [20]. Although, it can obtain extremely accurate solutions, the response performance is insufficient for problems with strict constraints.

With the above-described problems in mind, we propose a method of applying GA to solve the long-distance transportation problem. This solution method satisfies both interactive response performance and human-expert-level accuracy. The planning of the long-distance transportation network and its technical problems are described in section 2. In section 3, the concept of the method that meets the above requirements is proposed. In section 4, the detail implementation of the method is described. Section 5 shows the experimental results for VRP benchmark problems. In section 6, an experiment on the long-distance transportation problem is explained. Section 7 presents our conclusions.

II. Long-distance Transportation Network Planning Problem

A. Cooperative Logistics Network

Previously, parts suppliers individually delivered their own parts to factories. However, the cost of procurements logistics increased because of the increase in high-mix, low-volume production [21]. For this reason, to realize total optimization of the procurements network, enterprises that belong to a supply chain set up a cooperative logistics network using composed of depots (delivery center) as shown in Fig. 1.

This cooperative logistics network consists of several mutual sub-networks such as “parts-collection networks” covering parts suppliers and depots and a “long-distance transportation network” among factories and depots. The key features of the long-distance transportation network are

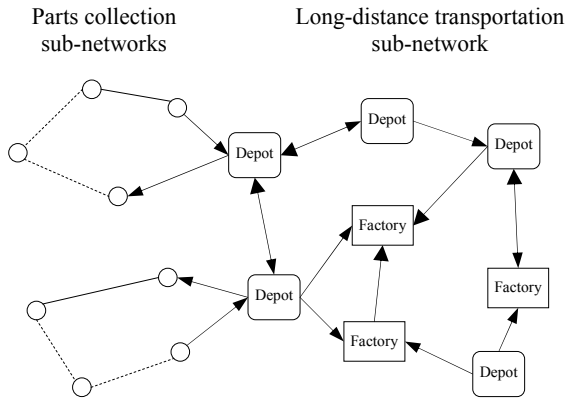


Figure 1. Structure of cooperative logistics network.

as follows.

(1) Wide-area transportation

Factories and depots are located across a large nationwide area. Long-distance transportation is thus necessary for this cooperative logistics.

(2) Strict time constraints

Transportation between depots and factories requires synchronization of the production plans of each factory. This parts transportation must therefore satisfy strict time constraints.

(3) Coexistence of pickup and delivery transportation

In this network, parts are transported from depots to factories. Moreover, semi-finished products are transported from one factory to another. Vehicles thus pick up parts and deliver parts simultaneously.

(4) Multi depots

There are several depots in the long-distance transportation network, and each depot has several vehicles. A vehicle starts from a depot and travels to several factories and other depots. Finally, the vehicle returns to the original depot. Along this traveling route, some parts are directly transported from a factory to other factories, that is, not through any depots.

B. Technical Problems Regarding Long-distance Transportation Network Planning

To implement the scheduling method for the long-distance transportation network, the following technical problems must be addressed.

(1) Human-expert-level optimality for strict-constraint problems

An optimal solution for solving the scheduling-phase problem is not always required. However, to avoid crucial error at estimation of cost and operability, at least, human expert-level optimality is required.

(2) Response performance for interactive operation

To realize a human expert’s evaluation from many aspects, response performance enabling interactive operation at least once or twice an hour is necessary.

C. Model for Long-distance Transportation Network

The scheduling problem is modeled, with the following

notations, as follows .
Q: maximum loading capacity of a vehicle.
M: maximum usable vehicles.
$\{L_i\}_{i=1..N}$: a set of loads.
$\{w_1, .., w_N, w_{N+1},.., w_{2N}\}$: a set of pickup and delivery works.
w_{2i-1}: pickup work of load L_i.
w_{2i}: delivery work of load L_i.
Q_i : amount of load that is picked up or delivered at work w_i.
When work w_i is a pickup work, Q_i is a positive value. On the contrary, Q_i is negative, when work w_i is a delivery. Q_i satisfies (1).

$$Q_{2i} = -Q_{2i-1}, Q_{2i-1} > 0, Q_{2i} < 0 . \quad (1)$$

$\{P_i\}_{i=1..P}$: a set of transportation points (e.g., factories and depots). And $Pl(w_i)$ means the transportation point where work w_i is done.
$[e_i, l_i]$: A time constraint for work w_i. A vehicle to which work w_i is assigned must arrive to delivery point $Pl(w_i)$ no later than l_i. When the vehicle arrives before e_i, it must wait the start of the work (pickup or delivery) until e_i.
M_i : number of vehicles that the distribution point P_i holds. M means the total of all M_i as (2).

$$\sum_{i=1}^{P} M_i = M . \quad (2)$$

A vehicle route is a sequence of works as follows.

$$W = \langle w_{i_1}, w_{i_2}, .., w_{i_n} \rangle \quad (Pl(w_{i_1}) = Pl(w_{i_n})) . \quad (3)$$

A vehicle starts from a depot and travels several points (factories and depots) to pick up or deliver parts. The vehicle finally comes back to the original depot. Thus, a traveling route of a vehicle is represented as a sequence of works (pickup or delivery).
W_k^i-: the set of works before w_k in the sequence.
W_k^i+: the set of works after work w_k in the sequence.
$Wrk(W_i)$: the set of works contained vehicle route W_i.
$WORK$: the set of all works.
$t(P_i,P_j)$: vehicle traveling time from points P_i to P_j.
$tm(W_i)$: total traveling time of vehicle route W_i.

$$tm(W_i) = \sum_{j=1}^{n-1} t(Pl(w_{i_j}), Pl(W_{i_{j+1}})) + t(Pl(w_{i_n}), Pl(W_{i_1})) . \quad (4)$$

$\omega = \{W_1, W_2, \ldots, W_m\}$: a transportation plan, where ω is a set of vehicle routes that satisfy constraints (5) and (6).

$$\bigcup_{i=1}^{m} Wrk(W_i) = WORK . \quad (5)$$

$$Wrk(W_i) \bigcap Wrk(W_j) = \varnothing (i \neq j) . \quad (6)$$

Moreover, a vehicle route W_i that is contained in ω must satisfy the following constraints.
(a) Limitation on number of vehicles.
The number of the vehicle that starts from the depot must not exceed the number of vehicles held by the depot.
(b) Time constraint
Let W_j be a vehicle route that work w_i is assigned to, and a_i^j be the arrival time of the vehicle to $Pl(w_i)$. The arrival time must not exceed time constraint l_i.

$$a_i^j \leq l_i . \quad (7)$$

(c) Vehicle-total-traveling-time constraint

$$tm(W_i) < T. \quad (8)$$

(d) Pick-up-and-delivery-work-order constraint
The corresponding pick-up work and delivery work must be assigned to the same vehicle by the following order (9).

$$w_{2j-1} \in W_i \rightarrow w_{2j} \in W_{2j-1}^i + . \quad (9)$$

(e) Load-capacity-constraint for a vehicle

$$0 \leq \sum_{k=1}^{j} Q_{ik} \leq Q \quad (1 \leq j \leq n) . \quad (10)$$

α: fixed cost for a vehicle.
$c(P_i, P_j)$: vehicle transportation cost from points P_i to P_j.
The vehicle traveling cost of route W_i is represented as (11).

$$tc(Wi) = \alpha + \sum_{j=1}^{n} c(Pl(w_{i_j}), Pl(w_{i_{j+1}})) + c(Pl(w_{i_n}), Pl(w_{i_1})) . \quad (11)$$

The total cost of transportation plan ω is thus represented as (12).

$$tc(\omega) = \sum_{i=1}^{m} tc(W_i) . \quad (12)$$

III. Concept of The Proposed Solution Methhod For The Cooperative Logistics Network Scheduling

A. *Concept of Selfish-constraint-satisfaction GA*

To guarantee both response performance and optimality in solving the transportation-planning problem as described in section 2, we propose a GA method, which we call "selfish-constraint-satisfaction GA". This method basically consists of the following two ideas.

The first idea is a new concept of a "selfish gene", which is different from Dawkins' "selfish gene". The original "selfish gene" proposed by R. Dawkins [22] 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; that is, 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 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, a divine being, namely, "God", or a "natural system" admits reproduction or prosperity of a chromosome or an individual, each constituent of which is a selfish gene. However, in the end, through reflection or punishment at some appropriate generations, only individuals that satisfy the objectives or constraints can survive over generations.

We came up with the above two ideas in order to find an efficient GA method that can optimally solve strict-constraint problems. Concretely speaking, in the GA method for solving strict-constraint transportation scheduling problems, each gene satisfies only the time constraints directly related to itself, in other words, it does not consider satisfying the time constraints of other genes in the same chromosome. Further, each such 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 is given through changing the structure of each constituting gene per execution of GA's operation such as crossover and mutation. 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, by allowing the selfishness of each constituting gene and not examining the global constraints' satisfaction except for some particular generations, GA's execution time is shortened. Moreover, by allowing various types of individuals to survive, more optimal solutions are found. Both high optimality and speedy responsiveness are thus obtained.

In the real world of living things, a group of unit genes or an individual that perfectly adapts to the environment does not suddenly emerge in their evolving stages. 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 each generation, the population size often shrinks or the system falls into a local minimum. Efficient optimization cannot be attained in this fashion. When constraints are very severe, in the worst case, the whole population can be destroyed. This is to say, if a population consists of only individuals that satisfy constraints, a remarkable improvement is not expected. However, there is a risk of falling into a local minimum. For such reasons, we also give a limited chance of survival, reflection or reparation to the individuals that violate constraints.

B. *Penalty for Time-constraint Violation and Reparation*

When we permit the existence of individuals that violate time constraints, it is natural that all individuals will eventually violate the time constraints. To avoid this, a penalty is added to the fitness value for an individual that violates the time constraints. Concretely speaking, the value in which the penalty coefficient is multiplied by the cost that corresponds to the delay time is added to the cost estimation.

Moreover, an individual that violates time constraints is repaired at constant generation intervals. To repair the time-constraint violations, pairs of pickup and delivery works on a vehicle route that violate the constraint are removed singly. After these removals are performed for all vehicle routes, the removed pickup and delivery works are reinserted into a vehicle route that globally satisfy time constraints. If there is no vehicle that can satisfy the time constraints, a new vehicle is allocated to the transportation.

C. *Dynamic Penalty Control*

The above mentioned penalty is regarded as latitude for individuals that violate time constraints. The large latitude promotes the convergence of the objective function. On the other hand, the small latitude prompts a search in the constraint satisfaction solutions. However, it is difficult to generate new individuals that satisfy all constraints for hard constraint problems.

To avoid this problem, we apply a dynamic penalty control method [23]. This method promotes the optimization of the objective function by using relatively large latitude at start time. Then, the latitude is decreased gradually in order to reduce the individuals that violate time constraints in the population. In more detail, the penalty is increased exponentially from initial value at each generation. At the reparation generation, the penalty is turned down to the initial value.

D. *Constraint Pre-checking Method*

To obtain highly optimal solutions by avoiding the convergence into the local minimum, the GA must have enough population and generations. However, long chromosomes containing many genes are needed to represent our planning problem. Therefore, both the mutation operation and the crossover operation take a long calculation time. Moreover, the calculation slows down because various constraints on our proposed GA are checked. To avoid such problems caused by various constraints, we propose a constraint pre-checking method. This method introduces into our GA the concept of constraint propagation utilized as an efficient solving method for constraint-satisfaction problems [24, 25]. Namely, this method statically analyses the constraints on the problem, before starting the GA. This reduces the overhead for dynamically searching nodes that cannot be visited by one vehicle. This improves the calculation efficiency of routes' reconstruction during the crossover and the mutation of GA. That is, constraints brought by, for example, distribution time range, load capacity, and node's geographical position are statically analyzed before starting the GA and set in a table of exclusive nodes. Thus, during the execution of GA, the system does not need to dynamically repeat the check of these constraints. In particular, because distribution points (factories and depots) are spread over a nation-wide area, the geographical conditions effectively reduce the combination of points that can be visited by one vehicle. Since there are many constraints on our long-distance transportation planning problem, it is expected that this method reduces the calculation time of the GA and realizes near-real-time response performance.

In the operation of our cooperative logistics network, a vehicle does not go to a far point (depot or factory) during running neighboring points. This operational rule is represented by the following constraints (13), (14), and (15). The x, y, and z represent transportation points. When these three condition are satisfied, the pickup or delivery work at

y cannot be inserted into a vehicle route between x and z.

$$l_0<t(x,z)<l_1. \quad (13)$$
$$t(x,y)+t(y,z)-t(x,z)>l_2. \quad (14)$$
$$t(x,y)+t(y,z)-t(x,z)>\beta t(x,z). \quad (15)$$

β,l_0, l_1, and l_2 are fixed values.

IV. Implementation Of The Proposed Method

A. Structure of Chromosome

The solution to the long-distance transportation problem includes clustering of loads, which indicate their correspondence to vehicles, and their travel routes in each cluster. To express both clustering and routing information, the chromosome structure shown in Fig. 2 (a) is applied. The chromosome is a sequence of work IDs of pickups and deliveries. Table 1 shows a detail of the load information. The load information consists of its origin, destination, pickup work ID, and delivery work ID. The "load 1" is transported from P1 to P5, and its pickup id is 1, and delivery work id is 2. In a similar manner, the "load 2" is transported from P3 to P4. In this example, the first vehicle of the chromosome in Fig. 2 (a) travels round P3, P4, P6, and P8 as shown in Fig. 2 (b).

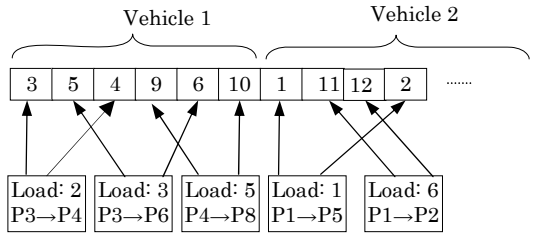
B. Scheduling Algorithm using GA

The selfish-constraint-satisfaction GA constructs solutions disregarding global constraints. This type GA generates individuals that violate time constraints in the population. Consequently, to solve this problem, these individuals have to be repaired at constant generation intervals. Moreover, dynamic penalty control is also applied for efficient optimization. Thus, in addition to the normal genetic operations such as selection, crossover, and mutation, reparation for individuals that violate time constraints and recalculation of the penalty are performed as follows.

Step 1. Generate initial population.
Step 2. Calculate penalties.
Step 3. Calculate fitness value for each individual.
Step 4. Perform selection.
Step 5. Perform crossover.
Step 6. Perform mutation.
Step 7. Repair individuals that violate time constraints.
Step 8. Check terminates condition.
Step 9. Go to step 2.

TABLE 1.
SAMPLE PICKUP AND DELIVERY WORK OF LOAD

ID	Origin	Destination	Pickup work ID	Delivery work ID
1	P1	P5	1	2
2	P3	P4	3	4
3	P3	P6	5	6
4	P5	P9	7	8
5	P4	P8	9	10
6	P1	P2	11	12



(a) Sample representation of chromosome

Vehicle 1: P3→P4→P6→P8
Vehicle 2: P1→P2→P5・・・

(b) Sample vehicle round routes

Figure 2. Sample representation of chromosome and vehicle routes.

C. Fitness Value and Selection

The fitness value of the individual becomes larger, when the total cost becomes smaller. To be more precise, let U is an enough large fixed value compared to the total cost, and the fitness value is defined as (16). To create a next generation, individuals are selected at a rate proportional to their fitness value. In addition, the elite individual who has the highest fitness value is always copy to the next generation.

$$U\text{-}tc(\omega). \quad (16)$$

D. Crossover and Mutation

To realize the above concept of selfish-constraint-satisfaction GA, an insertion method called "selfish NI (Nearest Insertion)" is applied to put nodes into a vehicle route. With this method, a node is inserted into a tour, using NI method, if the node satisfies its own constraints only. That is, the influence on the other nodes is not considered.

The crossover constructs a child from two parents according to the following process.

(1) Determine the crossover point in one parent chromosome.
(2) Obtain vehicle routes represented by a group of genes located before the crossover point in the chromosome.
(3) Change the order of remaining nodes that are not contained in any vehicle routes obtained in (2) according to the order of nodes (genes) in the other parent's chromosome.
(4) Insert the remaining nodes into the route obtained in (2) by using the selfish NI method in the order reordered in (3).
(5) If no vehicle satisfies the time and load-capacity constraints, a new vehicle is assigned to the transportation of the load.

In this crossover process, when a pickup node is deleted from a vehicle route, the corresponding delivery node is also removed. Namely, a pair consisting of a pickup work and a delivery work is deleted and inserted into the same

vehicle route simultaneously.

The mutation randomly selects a mutation node and deletes it together with its neighboring nodes. The deleted delivery points are reinserted into the transportation plan using the selfish NI method. This delete and reinsert process is done for each load. The pickup/delivery work pair is always assigned to the same vehicle.

E. Reparation

The selfish-constraint-satisfaction method permits to survive individuals who violate time constraints. Thus, these individuals should be repaired at constant generation intervals by the following two steps.

(1) Elimination of loads that violate time constraints

If a pickup or delivery time constraint is violated, both the pickup and delivery works are removed from a vehicle route. Accordingly, the chromosome shortens by this elimination.

(2) Reinsertion of eliminated load

The removed pickup and delivery works are reinserted into a vehicle route which satisfies constraints globally. When there is no vehicle which satisfies all constraints, a new vehicle is assigned to the schedule and the eliminated pickup and delivery works are inserted into the newly assigned vehicle. Namely, the pickup and delivery works are added to the end of the chromosome in Fig. 2 (a).

F. Proposed Solution Method

We propose the following two method using the above mentioned elemental solution methods.

(1) Selfish-constraint-satisfaction GA with fix penalty

This solution method constructs a vehicle route using the selfish NI method and permits the existence of individuals that violate constraints in the population. The penalty value is fixed for all generations in this method.

(2) Selfish-constraint-satisfaction GA with dynamic penalty control

This solution method also permits the existence of individuals that violates time constraints. Moreover, the penalty for individuals that violates time constraints is dynamically modified.

In the next section, experiments to evaluate the effect of the above two selfish-constraint-satisfaction GAs are performed. These tests compare them with Non-selfish GA. Non-selfish GA applies conventional non-selfish NI method to crossover and 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 of such positions of nodes are not searched out, crossover or mutation operation assigns a new vehicle.

V. EXPERIMENTS USING THE BENCHMARK PROBLEM

To evaluate the proposed selfish-constraint-satisfaction GA, we determined the penalty value for time-constraint violations, and the generation gap interval for the reparation using VRP benchmark problems [26]. We then compared the accuracy of the selfish-constraint-satisfaction GA with that of the global-constraint-satisfaction GA, which satisfies all constraints all the time, by these benchmark problems. In the next section, to evaluate the proposed method, we applied it to the long-distance-transportation-network planning problem.

GA parameters of population size, mutation rate, and crossover rate were determined based on an explorative experiment. In this experiment, we executed GAs in 30 minutes and compared solutions. As a result, we determined the population size, mutation rate, and crossover rate as 100, 5%, and 10% respectively.

A. Experiment Parameters

(1) Penalty coefficient for time-constraint violation

Our target long-distance transportation network contains about one hundred distribution points (factories and depots). Thus, we selected a VRP benchmark problem with 100 distribution points to determine the parameter for our proposed GA.

The accuracy of the solution according to the change in the coefficient is shown in Fig. 3. When there is no penalty, the error rate is over 10%. When the coefficient is larger than 3, the error rate stays under 5%. In particular, the error rate becomes less than 3% when a value from 5 to 10 is specified for the coefficient. According to this result, we determined the coefficient as 5.

(2) Generation gap for gene reparation

Fig. 4 shows the change in the accuracy according to the change in the generation gap interval of the time-constraint reparation. The error rate is over 3% when the gap is less than 3. The accuracy becomes a minimum when the gap is between 5 and 10. We thus determined the gap as 10 generations.

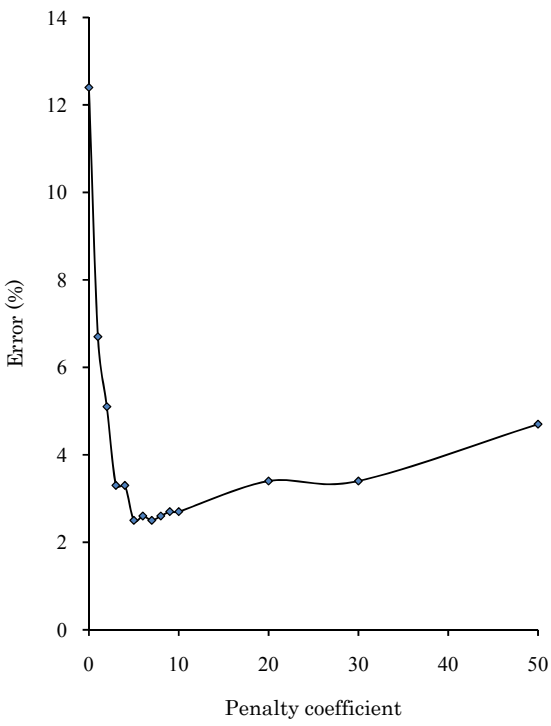


Figure 3. Correlation between penalty coefficient and accuracy.

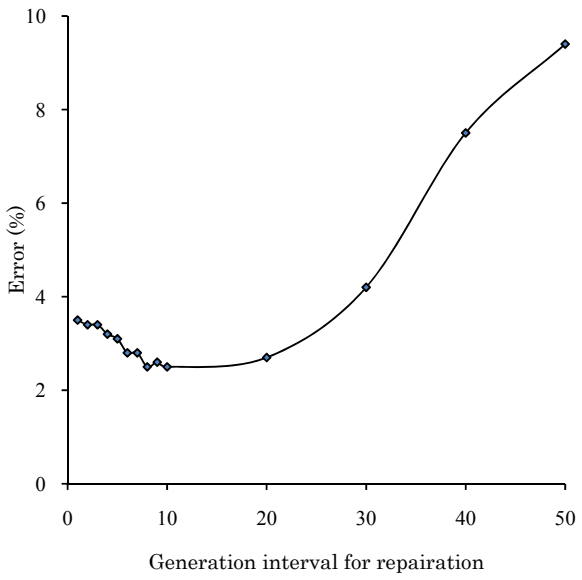


Figure 4. Correlation between generation gap and accuracy.

B. *Evaluation of Accuracy*

To evaluate the accuracy of the solution, we applied the proposed method to the VRP benchmark problems. Because of the tight connections between the production schedules of each factory, the long-distance transportation scheduling is assigned strict time constraints. We prepared three problems: "Problem 1", "Problem 2", and "Problem 3". Problem 1 is the bench mark problem (R101) which has the hardest time constraints in bench mark problems.

Problem 2 and Problem 3 have the same distribution points and load information as Problem 1. However, we reduced the time windows of Problem 2 to half of Problem 1. Moreover, Problem 3 has stricter time-constraints than Problem 2. The time windows of Problem 3 are reduced to half of Problem 2. Thus, the time windows of Problem 3 are one fourth of Problem 1. These three problems have the same optimal solution.

In this experiment, we solve the problem using the following three methods.

(1) Non-selfish GA
(2) Selfish-constraint-satisfaction GA with fix penalty
(3) Selfish-constraint-satisfaction GA with dynamic penalty control

Table 2 lists the accuracy of above three methods for the Problem 1 in calculation of 1000 generations (when the solution is converged). Though the accuracy of the non-selfish GA exceeded 4.5%, the selfish-constraint-satisfaction GAs can obtain the solution with under 3.0% error. Table 3 lists the accuracy for the Problem 2. The both selfish-constraint-satisfaction GAs obtains solutions with 2.7% error. Moreover, table 4 shows the results for the Problem 3. The selfish GAs can obtain the better solution than the non-selfish GA as the result of Problem 2. In particular, the third method can obtain the solution with under 4.0% error.

TABLE 2.
ACCURACY OF EACH METHOD FOR PROBLEM1

Method	Generation	Accuracy
Non-selfish GA	1,000	4.8%
Selfish-constraint-satisfaction GA with fix penalty	1,000	2.5%
Selfish-constraint-satisfaction GA with dynamic penalty control	1,000	2.5%

TABLE 3.
ACCURACY OF EACH METHOD FOR PROBLEM2

Method	Generation	Accuracy
Non-selfish GA	1,000	6.7%
Selfish-constraint-satisfaction GA with fix penalty	1,000	2.7%
Selfish-constraint-satisfaction GA with dynamic penalty control	1,000	2.7%

TABLE 4.
ACCURACY OF EACH METHOD FOR PROBLEM3

Method	Generation	Accuracy
Non-selfish GA	1,000	7.3%
Selfish-constraint-satisfaction GA with fix penalty	1,000	4.2%
Selfish-constraint-satisfaction GA with dynamic penalty control	1,000	3.8%

C. *Evaluation of Response Performance*

To evaluate the constraint pre-checking method that calculates the combination of points that the same vehicle cannot go around, we examined two methods: one uses constraint pre-checking and the other does not use pre-checking. The pre-checking method bumps up the generations that could be calculated within five minutes from 820 to 1045 as shown in table 5. Moreover, it also improves the accuracy from 2.9% to 2.6% error by the increase of the calculated generations.

VI. APPLICATION TO THE LONG-DISTANCE TRANSPORTATION PLANNING PROBLEM

To evaluate the proposed method using the same condition as the long-distance transportation problem, we prepared a transportation problem with mixed transportation (i.e., pickup and delivery) and many depots. This experimental network consists of 70 factories and 30 depots. Moreover, 500 loads were prepared for the network. We set the origin and destination, volume, and time constraints for each load. The constant values, l_0, l_1, l_2, and β in (13), (14), and (15) were set as 10km, 100km, 700km, and 7 respectively. Fig. 5 shows the convergence of the solutions.

The proposed methods can calculate 300 generations within 30 minutes. Table 6 shows the solutions of 300 generations GA. In this experiment the selfish-constraint satisfaction GA with fixed penalty can obtain about 10% less cost solution than the non-selfish GA. In addition, the cost is improved 0.5% due to the dynamic penalty control method.

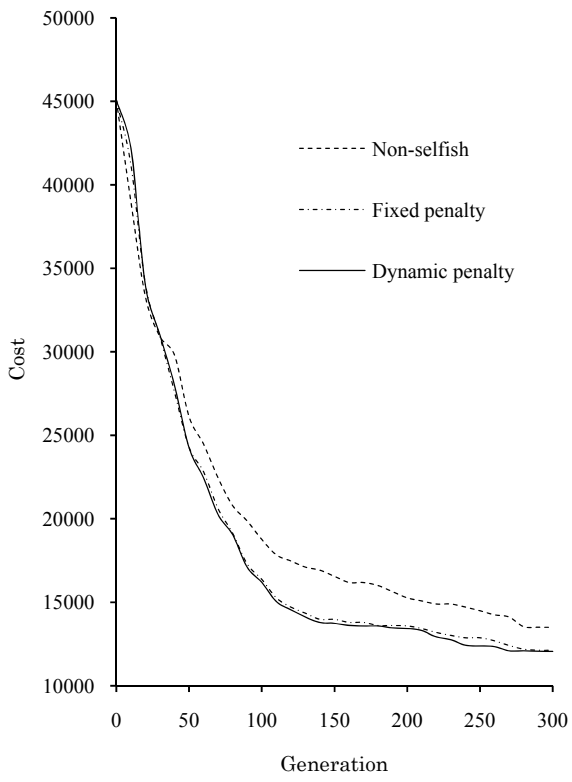


Figure 5. Convergence of solutions.

TABLE 5.
EFFECT OF CONSTRAINT PRE-CEHCKING METHOD

Method	Generation	Accuracy
Non pre-checking	820	2.9%
Pre-checking	1,045	2.6%

TABLE 6.
ACCURACY OF SOLVINE LONG-DISTANCE TRANSPORTATION PROBLEM

Method	Cost
Non-selfish GA	13,512
Selfish-constraint-satisfaction GA with fix penalty	12,132
Selfish-constraint-satisfaction GA with dynamic penalty control	12,065

VII. CONCLUSION

We proposed a planning method for a long-distance transportation network that obtains both interactive response performance and human expert-level accuracy. This solution method is based on the following three ideas.

1) First idea is selfish-constraint-satisfaction GA. Each gene, which constitutes an individual, does not care about constraints of all other genes within the same individual. Moreover, an individual is not killed until some particular generations even if it includes a selfish gene that cannot globally satisfy constraints. This method enables to avoid falling into a local minim for strict-constraint problems.

2) The second idea is the dynamic penalty control method. This method promotes the optimization of the objective function by increasing the penalty for the individuals that violate time constraints according to the reparation cycle.

3) The third idea is constraint pre-checking method. In our problems, the distribution points dispersed to a large nationwide area. Thus, this method statically analyses the constraints on the problem before starting the GA to reduce the overhead for dynamically searching nodes that cannot be visited by one vehicle.

Our experimental results revealed that our proposed method enables interactive operations once or twice an hour for problems on a practical scale. Moreover, this method can obtain solutions with human expert-level accuracy for hard constraint problems.

REFERENCES

[1] T. P. Stank and P. J. Daugherty: "The impact of operating environment on the formation of cooperative logistics relationships", in *Transportation Research Part E: Logistics and Transportation Review*, Vol. 33, No. 1, pp. 53-65, March 1997.

[2] M. Kubo, *Logistics engineering*, Tokyo: Asakura-Syoten, 2001.

[3] M. Wasner and G. Zapfel: "An integrated multi-depot hub-location vehicle routing model for network planning of parcel service", in *Int'l Journal of Production Economics*, Vol. 90, No. 3, pp. 403-419, Aug. 2004.

[4] T. Onoyama, S. Kubota, K. Oyanagi, and S. Tsuruta, "A method for solving nested combinatorial optimization problems –a case of optimizing a large-scale distribution network–", in *Proc. IEEE SMC 2000*, pp. 2467-2472, Oct. 2000.

[5] T. Onoyama, T. Maekawa, S. Kubota, and N. Komoda, "High-speed planning method for cooperative logistics networks using mixed integer programming model and dummy load", in *Trans. Electrical Engineering in Japan*, Vol. 164, No. 2, pp. 64-70, Feb. 2008 [in *Trans. IEEJ*, Vol. 127-C, No. 1, pp.31-36, Jan. 2007].

[6] H. J. Sebastian: "Strategic planning of distribution networks for letter and parcel mail using optimization models", in *Abstract of the 19th Int'l Workshop on Complex Systems Modeling jointly with the 6th Int'l Symposium on Knowledge and Systems Sciences (CSM/KSS'05)*, pp. 20-21, Aug. 2005.

[7] J. Bramel and D. Simchi-Levi: *The Logic of Logistics*, New York: Springer-Verlag (1997).

[8] M. Gendean, G. Laporte, and J.Y. Potvin, "Vehicle routing modern heuristics", in E. Aarts and J. K. Lenstra (Eds.), *Local Search in Combinatorial Optimization* , Princeton: Princeton University Press, pp. 311-336, 1997.

[9] T. Onoyama, T. Maekawa, S. Kubota, S. Tsuruta, and N. Komoda, "Vehicle routing problem solving method for a cooperative logistics network by using multi-stage GA", in *Trans. IEEJ*, Vol. 127-C, No. 9, pp. 1460-1467, Sep. 2007.

[10] P. Toth and D. Vigo: "Exact solution of the vehicle routing problem", in T.G Cranic and G. Laporte (Eds.), *Fleet Management and Logistics*, Kluwer Academic Publishers, pp. 1-31, 1998.

[11] T. Masui, S. Yurimoto, and N. Katayama, *OR in Logistics*, Tokyo: Makisyoten, 1998.

[12] H. Mohri, M. Kubo, and M. Mori, "Distribution planning problem on mainline transportation", in *Abstract of Japan OR Society 1995*, pp. 128-129, Sep. 1995.
[13] S. Ataka and M. Gen, "Solution method of multi-product two-stage logistics problem with constraints of delivery course", in *Trans. IEEJ*, Vol. 128, No. 3, pp. 456-461, March 2008.
[14] H. Mohri, T. Watabnabe, M. Mori, and M. Kubo: "Cost allocation on vehicle routing problem", in *Journal of the Operations Research Society of Japan*, Vol. 40, No. 4, pp. 451-465, Jan. 1997.
[15] S. M. Sait and H. Youssef: *Iterative Computer Algorithms with Applications in Engineering: Solving Combinatorial Optimization Problems*, Los Alamitos: IEEE Computer Society, 2000.
[16] N. Park, H. Okano, and H. Imai, "A path-exchange-type local search algorithm for vehicle routing and its efficient search strategy", in *Journal of the Operations Research Society of Japan*, Vol. 43, No. 1, pp. 197-208, Jan. 2000.
[17] E. Taillard, P. Badeau, F. Guertin, and J.-Y. Potvin: "A tabu search heuristic for the vehicle routing problem with soft time windows", in *Transportation Science*, Vol. 31, No. 2, pp. 170-186, Feb. 1997.
[18] M. Gendreau, H. Alain, and G. Laporte: "A tabu search heuristics for the vehicle routing problem", in *Management Science*, Vol. 40, No. 10, pp. 1276-1290, Oct. 1994.
[19] J. H. Hooker and N. R. Natraj: "Solving a general routing and scheduling problem by chain decomposition and tabu search", in *Transportation Science*, Vol. 29, No. 1, pp. 30-44, Jan. 1995.
[20] G. A. P. Kindervater and M. W. P. Savelsbergh: "Vehicle routing: handling edge exchanges", in E. Aarts and J. K Lenstra (Eds.), *Local Search in Combinatorial Optimization*, Princeton University Press, pp.337-360, 1997.
[21] H. Kuse: "Logistics and social problems", in *Journal of the Society of Instrument and Control Engineers*", Vol. 37, No. 3, pp. 166-169, March 1998.
[22] R. Dawkins, *The Selfish Gene*, London: Oxford University press, 1989.
[23] Y. Sakurai, T. Onoyama, S. Kubota, Y. Nakamura, and S. Tsuruta, "Locally selfish-gene tolerant dynamic control GA to solve constraint TSPs", in *Trans. IPSJ*, Vol. 48, No. SIG-19, pp. 127-138, Dec. 2007.
[24] V. Kumar: "Algorithms for constraints satisfaction problems: A survey", in *AI Magazine*, Vol. 13, No. 1, pp. 32-44, Jan. 1992.
[25] P. Shaw: *Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems*, Berlin: Springer-Verlag, 1998.
[26] M. M. Solomon: VRPTW benchmark problems. http://web.cba.neu.edu/~msolomon/problems.htm

Takashi Onoyama was born in 1959. He received the Bachelor of Science from the Osaka University in 1981. He received the PhD in Information Science from the Osaka University in 2007.

He joined Hitachi Software Engineering Co., Ltd in Tokyo, Japan in 1981. In 2003, he became Department Manager of Research & Development Department. The focus of his research is concentrated within the topics of planning method for logistics system.

Dr. Onoyama is a member of the IEEE and others.

Takuya Maekawa was born in 1971. He received the Bachelor of Science from the Yamagata University in Japan in 1993.

He joined Hitachi Software Engineering Co., Ltd in 1993 and he has been developed scheduling system. His main research interests are VRP solving method for logistics planning.

Mr. Maekawa is a member of the IPSJ.

Yoshitaka Sakurai was born in 1977. He received Bachelor of Information Science from the University of Electro-Communications in 2000. He received the PhD in Engineering from the University of Electro-Communications in 2005.

He teaches at the Tokyo Denki University from 2005. His main research interest is theory and application of evolutional and learning algorithms.

Dr. Sakurai is a member of the IEEE and IPSJ.

Setsuo Tsuruta was born in 1947. He received the Bachelor from the Waseda University in 1971 and he completed the Master of Engineering at the Nagoya University. He received the PhD in Engineering from the Nagoya University.

He joined Hitachi Ltd in 1973. In 2003, he joined the information environment faculty of Tokyo Denki University as a professor. His research focuses on the scheduling system and intelligent system.

Prof. Tsuruta is member of IEEE and others.

Norihisa Komoda was born in 1950. He completed the Master of Engineering at the Osaka University in 1974. He received the PhD in Engineering from the Osaka University in 1981.

He joined Hitachi Ltd in 1974. In 1991 he joined the engineering faculty of Osaka University as associate professor. Since 1992 he is professor of Graduate School of Information Science and Technology of the Osaka University. He is mainly interested in the information system in the area of manufacturing and distribution industry and knowledge based information processing.

Prof. Komoda is a member of IEEE and others. He received 1998 IEEJ Paper Award.