A General Framework of Resource Allocation Optimization and Dynamic Scheduling

Shuang Chang *, Manabu Ichikawa **, Hiroshi Deguchi *, and Yasuhiro Kanatani **

Abstract: Resource allocation and scheduling under scarce resources and limited time are always critical and challenging tasks, not only because of the complex situation with diverse needs involved, but also of any unpredictable occurrence during the whole dynamic process. This work proposes an agent-based framework to integrate the resource allocation and scheduling under a set of limitations, which could respond to contingent changes as a dynamic system. We focus on the following research questions and formulate them as a constraint satisfaction problem: how many resources should be assigned and dispatched to which location, in which sequence and under what process scheduling with time, resource availability, and ability-matching limits. We first give the corresponding formal definition, and then combine real-coded genetic algorithm and dynamic scheduling of multi-functional resource assignment to tackle the above proposed research questions. In addition, we experiment the model with a small make-up case to suggest some preliminary scenarios. In future, this framework would be further applied to real life emergency situations with empirical data for training purposes and providing insight for relevant policy makers.

Key Words : resource allocation, dynamic scheduling, agent-based approach.

1. Introduction

Efficient and effective resource allocation and scheduling under changing situations are challenging issues. The resource mentioned here is in a broad sense. It is not limited to a material resource, but may also refer to a personal resource. Limited time, heterogeneous stakeholders with diverse needs and resources, unexpected occurrence during the dynamic process, changing availability of scarce resources, and dependencies between the needs and available resources make this non-linear problem even more complicated.

In operational research and disaster management, this kind of problems are primarily conceptualized as resource allocation optimization problems [1]–[3] or operation scheduling problems [4]–[6] separately aiming to facilitate the plan-making and implementation in an efficient and effective manner. Especially, some of the related works focused on particular perspectives of the applications, such as vehicle dispatching and hospital patient scheduling [7]. However, in many circumstances, resource allocation and scheduling are closely related with each other as a dynamic system and will influence the operations of each other. In other words, they are not one-time only activities and operations, but continuously executing ones. For instance, when transporting injured patients to hospitals during a disaster, the transportation plan may continuously depend on the availability of doctors and facilities of each hospital, which would be based on the in-hospital resource scheduling changing along with the number of incoming patients. To the best of our knowledge, few of the existant works provided a holistic framework to capture such resource allocation optimization and in-location scheduling as a series of actions from the starting place to the final destination, considering a set of constraints as ability-matching, time-distance limitations and resource scarcity. Furthermore, responding to a continuously changing emergent environment, some of the works handled any unexpected occurrence as uncertainties [8], whilst we aim to reflect any allocation and scheduling changes due to the changing environment in a more straight-forward way to enable real-time support. Therefore, we will propose a general framework to integrate the resource allocation optimization and dynamic scheduling to respond to a changing environment in a time-efficient and demand-supply matching manner. Moreover, we formulate this problem as a constraint satisfaction problem, which has been applied in some resource allocation problems [9]. Within this framework, the changing availability of resources due to the scheduling will further influence the resource allocation subsequently as a dynamic and holistic process.

On the other hand, in order to optimize the resource allocation and scheduling, evolutionary algorithms, such as genetic algorithm [10] and simulated annealing [11], as well as various scheduling techniques have been widely applied [8],[12]. In addition, stochastic techniques are also applied extensively in operational management, especially in the disaster management realm, and interested readers could refer to [13] for a survey of this research stream. In this work, in order to catch the heterogeneity of needs and resources, and to cope with the dynamic and complex process, we will propose an agent-based approach integrating Ono’s real-coded genetic algorithm [14], and Deguchi’s dynamic scheduling of multi-functional resource assignment [15] to solve the following research problems: how many resources should be assigned to which location with ability-matching limits, and how is the resource scheduling processed within each location.

This general framework indicates various potential applications. Taking disaster management as an example, it could be applied to determine how to transport injured patients with dif-
different symptoms during a disaster to the corresponding hospitals in which they could be cured while minimizing the transportation time. Subsequently, based on the results we could determine how to schedule the transportation facilities as well as the in-hospital resources, such as doctors, nurses and beds. Another example could be allocating the required post-disaster resources, such as food, water, and tents to temporary shelters and then making the in-location schedules based on pre-defined activity paths.

This paper is structured as follows. We formulate the research problem as a constraint satisfaction problem and formalize an agent-based framework integrating real-coded genetic algorithm and dynamic scheduling in Section 2. We simulate the proposed agent-based models and discuss some preliminary evaluation results in Section 3. The conclusion with some future work will be presented and discussed in Section 4.

2. Modeling

The general framework is composed of two major stages. The first stage considers how many resources should be dispatched to which location under a set of constraints to minimize the gap between estimated needs and the actual allocation, and then making the in-location schedules based on pre-defined activity paths.

The detailed framework is shown in Fig. 1.

2.1 Constraint Satisfaction Problem

A constraint satisfaction problem (CSP) [16] could be formulated as follows:

1. A finite set of variables, \( X = \{X_1, X_2, \ldots, X_n\} \);
2. A domain set \( D \), containing a finite and discrete domain for each variable \( X_i; D = \{D_1, D_2, \ldots, D_n\}; D_i = \{d_1, d_2, \ldots, d_m\} \);
3. A constraint set, \( C = \{C_1, C_2, \ldots, C_m\} \), where \( C_i \) denotes a combination of values that a subset of variables in \( X \) can take.

Thus, the search space of a CSP, \( S \), is a Cartesian product of the \( n \) sets of finite domains, namely, \( S = D_1 \times D_2 \times \cdots \times D_n \). A solution for a CSP, \( s \in S \), is an assignment to all variables such that the values satisfying all constraints.

2.2 Model Formulation

In this sub-section, we give a formal definition of the proposed agent-based framework. Basically we define two types of agent: an agent with certain resource/ability and an agent with certain need. In addition, we require that each agent must be at one location at one time unit. The movement of agents among locations is two-way, which means the agent with resource could move towards the agent with need as satisfying the need, whilst the agent with need could move towards the agent with resource as well. Taking the examples of disaster management again, moving emergency resources to shelters is moving agents with resources to agents with needs, whilst moving patients to corresponding hospitals is moving agents with needs to agents with resources. The formal model definition is provided in the following subsections.

2.2.1 Agent and Spot

**Location.** We define a set of locations as \( L = \{l_1, l_2, \ldots, l_m\} \), of which each location \( l_i \in L \) is defined by \( (X_i, Y_i) \) as the coordinate, i.e. latitude and longitude respectively. This location could be hospital, disaster occurred spot, and relief base, etc.

**Agent with ability.** We define a set of agents with certain resource as \( P = \{p_1, p_2, \ldots, p_n\} \), of which each agent possesses certain ability to perform corresponding tasks or satisfying certain needs, such as doctors, rescue teams, required material resources, etc. Each \( p_i \in P \) is defined by its functional ability \( ability_i = Ability = \{a_1, a_2, \ldots, a_k\}, k \in N \).

**Agent with need.** We define a set of agents with certain need as \( Q = \{q_1, q_2, \ldots, q_n\} \), of which each \( q_i \in Q \) possesses a need \( need_i = Need = \{ne_{i1}, ne_{i2}, \ldots, ne_{ik}\}, k \in N \). This need could be resource or personnel, for example, injured people needs doctors (agents with ability).

2.2.2 Agent at Spot

Each agent \( p_j \in P \) stays at one location \( l_i \in L \) at each time unit, which makes this location possess a set of abilities as \( Ability_i \subseteq Ability \), indicating that the ability of location \( l_i \) includes the abilities of all agents currently staying at this location, and the capacity of each available ability at this location is defined as \( Capacity_i^{sup} = \{c_1, c_2, \ldots, c_k \mid c_i \in N\}; c_i \in Capacity_i^{sup} \) is the number of agents with ability \( a_i \in Ability_i \) currently at location \( l_i \).

Similarly, each agent \( q_j \in Q \) stays at one location \( l_i \in L \) at each time unit as well, which makes this location possess a set of needs as \( Need_i \subseteq Need \), indicating that the needs of location \( l_i \) include the needs of all agents currently at this location. The quantity of every need at this location is defined as \( Capacity_i^{need} = \{c_1, c_2, \ldots, c_k \mid c_i \in N\}; c_i \in Capacity_i^{need} \) is the number of agents with need \( ne_i \in Need_i \) currently at location \( l_i \).

Here we define a bijective mapping function \( Mapping : Ability \rightarrow Need; Mapping(a_i) = ne_i \), indicating that \( a_i \in Ability \) could, and only could satisfy one need \( ne_i \in Need \), and vice versa.
2.2.3 Agent moving to Spot

Both agent \( p_i \) and agent \( q_j \), with need \( n_e_i \), possess a variable list \( X_i = \{x^i_j\} \), \( l_j \) s.t. \( n_e_i \in \text{Need} \) or \( a_j \in \text{Ability} \), as the probability of going to a set of ability-matching locations, i.e. \( a_j = \text{Mapping}^{-1}(n_e_i) \) and \( \text{Mapping}(a_j) = n_e_i \). This probability could be interpreted at both micro level (as the probability of each agent) and macro level (as the proportion of agents with the same ability currently at the same location).

2.2.4 Constraint satisfaction problem

Depending on the situation, each agent \( q_j \in Q \) with need \( n_e_i \in \text{Need} \) would choose a location \( l_j \) as the destination where \( a_j \in \text{Ability} \), \( a_j = \text{Mapping}^{-1}(n_e_i) \), or each agent \( p_i \) with ability \( a_j \in \text{Ability} \) would choose a location \( l_j \) where \( n_e_i \in \text{Need}_j \), \( n_e_i = \text{Mapping}(a_j) \). The CSP is thus defined as follows.

1. Given a finite set of variables \( X = \{X_i\}, i \in \{1, 2, \ldots, n\} \)
2. A discrete and finite domain set \( D = \{D_1, D_2, \ldots, D_n\} \) is defined as \( D_j = \{\text{PROB}(l_i)\ldots \text{PROB}(l_j)\}\), \( l_j \in L \), where \( \text{PROB} : L \rightarrow [0, 1] \) is a probability function indicating the probability of choosing a location.
3. A constraint set \( C = \{C_j(X_i)\} \), where \( C_j(X_i) = \{l_j\} \), \( l_j \in L \) s.t. \( a_i \in \text{Ability}_j \); \( C_j(X_i) \) as the number of possible locations for this agent, and \( \sum_{a_i \in \text{Ability}_j} \text{PROB}(l_i) = 1 \).

The set of possible solutions is then defined as the Cartesian product of the sets of domains.

For the first resource allocation stage, we aim to minimize the time cost of movement and the difference between expected and actual allocation while satisfying a set of ability-matching constraints. The decision variables would be \( X_i = \{x^i_j\} \) of each agent. Formal formulation of the optimization purpose and constraints, and detailed explanation of the parameters will be introduced in the following subsection.

2.3 Real-Coded Genetic Algorithm

We apply Ono’s real-coded genetic algorithm [14] to solve this constraint satisfaction problem as the first stage of the framework, aiming to minimize the time cost and the difference between the expected and actual allocation while satisfying a set of ability-matching constraints. The reason why we choose this genetic algorithm is that its crossover operation, namely unimodal normal distribution crossover (UNDX), enables more efficient optimization on functions when epistasis among parameters are considered, no matter whether the function is unimodal or multimodal. In this sense, this genetic algorithm is very suitable for our application since there are dependencies among the decision variables of our fitness function. This function has a real number vector representation and it follows the minimal generation gap model (MGG) [17] for the population generation. The genetic algorithm is formulated as follows,

### Chromosomal representation

The length of the chromosome is the total number of possible locations moving all types of agents. For example, if there are agents with 3 types of ability or need at one location going to be dispatched, and the possible location set for each type of agent is \( \{l_1, l_2, l_3, l_4, l_5\} \), \( \{l_2, l_3, l_4\} \), and \( \{l_1, l_5, l_6\} \), respectively, then the length of the chromosome will be 10. If the agents are currently stay in 2 locations, then the length will be 20. Each unit of the chromosome \( x_{ij} \) represents the probability of assigning this agent to the destination \( l_j \) satisfying the constraint \( C_j(X_i) \), ranging between 0 and 1. By making this chromosomal representation, the hard constraint of ability-matching is satisfied naturally.

### Fitness function

The objective is to minimize the total time cost \( \sum\sum x_{ij} \cdot \text{Distance}_{e_j} \) and the difference between the expected and actual allocation \( \sum\sum |(c_{ij} - x_{ij} + N_j)| \). The fitness function \( F(X) \) is defined as follows,

\[
F(X) = \min(w_1 \sum\sum x_{ij} \cdot \text{Distance}_{e_j} + w_2 \sum\sum |(c_{ij} - x_{ij} + N_j)|) \tag{1}
\]

Where \( x_{ij} \) is agent \( q_j \’s \) (or \( p_i \’s \) probability of going to position \( l_j \), and \( \text{Distance}_{e_j} \) is the actual shortest distance between location \( l_j \) (destination) and \( l_h \) (current location). The distance could be calculated by Dijkstra’s algorithm and implemented by osm2po (an open source java package, ver. 5.0) [18]. \( c_{ij} \) is the capacity of ability \( a_i \) at location \( l_j \), \( N_j \) is the total number of agents with need \( n_e_i \). \( w_1 \) and \( w_2 \) are weights to adjust the proportion of two objectives, the values of which are generated randomly following Murata et al. [19]. \( \sum\sum x_{ij} \cdot \text{Distance}_{e_j} \) (as indicator 2) and \( \sum\sum |(c_{ij} - x_{ij} + N_j)| \) (as indicator 1) are normalized as \( v'_j = (v_j - v_{\text{min}})/(v_{\text{max}} - v_{\text{min}}) \) before the calculation.

### Constraints

There are two constraints. One is the hard ability-matching constraint, and the other is to make sure every agent is allocated to one and only one location. The formulations are as follows,

\[
a_i = \text{Mapping}^{-1}(n_e_i) \in \text{Ability}_j \tag{2}
\]

\[
\sum_j x_{ij} = 1, \text{for } \forall n_e_i \in \text{Need} \tag{3}
\]

Where \( a_i \) is the ability matched with need \( n_e_i \) of agent \( q_j \) and \( \text{Ability}_j \) is the to-be-distributed location \( l_j \’s \) ability set. The way to guarantee the second constraint is to set \( x_{ij} = \sum_j x_{ij} \) for any \( l_j \) such that \( a_i \in \text{Ability}_j \).

By designing the chromosome in this way, the length of each individual only depends on the multiplication of the number of ability types and possible locations, not the number of agents with different abilities. In the case of allocating a significant number of agents, the computation complexity of this real-coded GA design will outperform those of which the chromosomal length depends on the number of agents, with equal number of population and similar mutation/crossover operators. In this sense, this framework is capable to be applied to large-scale applications, especially to those with a significant number of agents and a relatively moderate number of ability types and locations. On the other hand, in order to handle extremely large-scale applications with a significant number of locations and ability-need pairs, we could adopt Ono’s [20] improved version of grid-oriented genetic algorithm to guarantee a speedy computation.

2.4 Dynamic Scheduling

The above real-coded genetic algorithm is applied to decide which location to go in order to minimize the time cost
and the difference between the expected and actual allocation. Based on its simulated results, we apply a revised version of Deguchi’s dynamic scheduling of multi-functional resource assignment [15] for the actual movement and inner-location resource scheduling with limited resources in terms of transportation facilities and in-location resources. This algorithm provides a simple yet efficient resource scheduling especially for handling parallel tasks with multi-function resources naturally. In addition, it allows any change of the capacity of available resources during any time and reflects the resulted schedules in time, which is very critical for handling changing situations. Furthermore, the resulted changes of the capacity of available resources or needs due to re-scheduling will be updated accordingly and applied to the first stage in the following cycle. In this sense, the changes during the scheduling process could be captured immediately and the results of which are reflected within this holistic framework as a dynamic system.

We treat each agent as a project, and the movement and in-location resource scheduling as a series of tasks. The corresponding formal definitions following Deguchi [15] are given as follows.

**Task Set.** $PS = \{q_1, q_2, \ldots, q_n\}$ is a set of unit projects. Here one agent $q_i \in Q$ is represented as one project. It could be $\{p_1, \ldots, p_m\}$, $p_i \in P$ as well and all the following definitions apply since the movement is two-way. Without losing generality, we only use $q_i$ in the following definitions. $TASKS[q_1] = TASKS[q_2] = \cdots = TASKS[q_n] = \{M_{ij}, H_{ij}\}$ is the task set for each project, $M_{ij}$ indicates the actual movement from departure spot $l_i \in L$ to destination $l_j \in L$, and $H_{ij}$ is the in-location scheduling of resources with ability $a_i \in Ability_j$ at location $l_j \in L$.

**Professional Set.** $PROFS = \{prof_{ij}, transport_1, transport_2, staff_j, facility_j\}$ is a set of professions. $prof_{ij}$ indicates the agents $p_i \in P$ with ability $a_i \in Ability_j$ at location $l_j$; $transport_1$ and $transport_2$ represent transportation facilities, such as ambulance and helicopter; $staff_j$ represents auxiliaries at location $l_j$; $facility_j$ represents location $l_j$ facilities, such as available beds in disaster management applications.

**Task Profession Relation.** The relation among Task, Profession, and the available resources is illustrated in Fig. 2.

---

**Profession assignment function.** For each profession, there is a set of available resources defined by this function, $ProfAssignF(w_{ij}) = transport_1$; $ProfAssignF(w_{2j}) = transport_2$; $ProfAssignF(w_{a_{ij}}) = prof_{ij}$, $w_{a_{ij}} \in P$ with $a_i \in Ability_j$ at location $l_j$; $ProfAssignF(w_{dj}) = staff_j$; $ProfAssignF(w_{f_{ij}}) = facility_j$. The resources could be either personnel or mere facilities.

**Possible task set for w.** $POSTASKS[w]$ is defined as the possible tasks which could be performed by each resource. $POSTASKS[w_{1j}] = POSTASKS[w_{2j}] = \{M_{ij}\}$; $POSTASKS[w_{a_{ij}}] = POSTASKS[w_{s_{ij}}] = POSTASKS[w_{f_{ij}}] = \{H_{ij}\}$.

**Estimated Time for Tasks.** $EstTimeAccompF$ is the function to assign required time unit for completing each task. $EstTimeAccompF(H_{ij})$ will be determined by assumptions and $EstTimeAccompF(M_{ij})$ is proportional to the real distance between the departure spot and the destination.

**Path Definition.** The path for each project is defined as $path_{ij} : start \rightarrow M_{ij} \rightarrow H_{ij} \rightarrow end$ depending on the departure location $l_i$, the destination location $l_j$ which is resolved by the genetic algorithm at the first stage, and the ability $a_i \in Ability_j$.

When assigning resources to an unfinished tasks set as a partial ordering, the tasks that request fewer resources will be given a higher priority and completed first. In addition, different from [15] in which unfinished tasks with shorter time period will be given a higher priority to compete for the resources, in this work the resources will be assigned randomly to unfinished tasks which require the same amount of resources. The input and output files of the dynamic scheduling are illustrated in Fig. 3.

### 3. Simulation

In this section, we set up a small-scale case to simulate the framework for demonstration purposes.

#### 3.1 Setting of Agents

We assume there are 500 agents at 5 locations with different needs $need_i \in \{a, b, c, d, e\}$, and 88 agents at 4 locations with corresponding abilities $ability_i \in \{a, b, c, d, e\}$. Those agents with needs will be allocated to locations where agents with abilities exist. The detailed information is listed in Tables 1 and 2. In addition, the distance between locations is estimated based on real GIS data via Google Map [21] and listed in Table 3.

#### 3.2 Setting of Real-Coded GA and Dynamic Scheduling

We set the real-coded GA parameters following Ono’s setting [14]: the number of crossovers for MGG: 100; population size: 500; $\alpha$ and $\beta$ for UNDX: 0.5 and 0.35 respectively; the number of generations: 2000. We run the simulation for
Table 1 Agents with abilities.

| Location | Coordinate     | Abilities | Capacities |
|----------|----------------|-----------|------------|
| l1       | 35.894444;139.623056 | a,b,c,d,e | 10.5,5,7.3 |
| l2       | 35.890930;139.679232 | c,d       | 5,10       |
| l3       | 35.890444;139.703056 | a,c,e     | 5,10,9     |
| l4       | 35.84325;139.730556 | a,e       | 8,11       |

10 times, and the result with the least evaluation value will be passed to the dynamic scheduling phase.

The simulation setting of the dynamic scheduling phase is as follows: each agent is treated as a project, and depending on their original location, destination and needs, the path \(\text{path}_{ij} : M_j \rightarrow H_j\) of the scheduling will be determined accordingly. The value of other resources is shown in Table 4.

We implemented the model by the programming language Java and executed it on a MacBook Pro equipped with processor Intel Core i7 (2.5 GHz), 16 GB RAM, and operating system OS X 10.10. The processing time of 10 runs of the simulation under the above-mentioned simulation parameter setting was around 23 seconds.

### 3.3 Simulation Result

Figure 4 (a), (b) and (c) show the landscape results of the probability of moving agents to each possible location of all 20 runs, by indicating the maximum, minimum and average value. x-axis is the 60 possible allocation cases, and y-axis is the corresponding probability. The order of these 60 possible allocation cases could be illustrated by using Table 5. For instance, case 1, 2, and 3 are of the probability of moving agents with needs \(a\) at location \(l_3\) to location \(l_1, l_2\) respectively, case 4 is of the probability of moving agents with needs \(b\) at location \(l_2\) to location \(l_1, l_2\) respectively, case 5, 6, and 7 are probabilities of moving agents with needs \(c\) at location \(l_2\) to location \(l_1, l_2\) respectively, and so forth. Since for each departure location there are 12 cases of all location, and there are 5 such departure locations, the number of total case is thus 60. Figure 4 (d) shows the average evaluation value of 10 runs along 2000 evaluation iterations. A snapshot of the allocation result with the best evaluation value of 10 runs when both indicators are considered is shown in Table 5. The probabilities shown below are rounded up to 2 decimal places for demonstration purpose, thus may not sum up to 1 due to numerical errors. To calculate the number of agents, we could simply multiply the probability with the number of agents of each need. The results shown in this table imply an effective resource allocation in the sense that all the agents are transported to the locations where their needs could be satisfied while optimizing certain criteria.

Figure 5 shows the evaluation value of both indicators of the best 50 children in the last 1000 evaluation iterations of real-coded GA. For indicator 1, the difference between the expected and actual allocation, x-axis is the evaluation value composing both indicators and y-axis is the evaluation value of the difference. For indicator 2, the distance of movement, x-axis is the evaluation value composing both indicators, whilst y-axis is the evaluation value of the distance. Figure 5 (a), (b), (c) show respectively the situation that only indicator 1, indicator 2 and both indicators are optimized. In Fig. 5 (d), x-axis and y-axis represent the evaluation value of indicator 1 and 2 respectively, from which we could observe that both indicators are minimized.

Regarding the dynamic scheduling, it will derive three tables from which we could check the assigned resources and starting time of each project (agent) (as in Fig. 6 (a) and (c)). In addition, from the perspective of resources, we could know their schedule in terms of assigned project along the process (as in Fig. 6 (b)). By analyzing the tables, we could identify the bottleneck of resource allocation, i.e. the lack of critical resources.
and adjust the initial resource setting accordingly. In addition, we could estimate the availability of each type of resource at any given time based on the scheduling tables, and then apply the estimated availability for any further resource allocation optimization. Table 6 shows the maximum, minimum and average busy time units of each type of resource, out of the whole scheduling process time units, 96. Because we assume all the resources are available at all times, we could identify the abundant resources which should be eliminated. For cases of which resources have working limits, we could specify them in the input table and examine the effects intuitively from the output tables.

In future work, we could further revise the scheduling algorithm according to different priority settings of tasks to improve the performance and make it more flexible to various scenarios. The results could be visualized to facilitate the policy-making process among stakeholders.

4. Conclusion

In order to achieve efficient and effective resource allocation and scheduling as a holistic process under ever-changing emergency circumstances, this work provided a general dy-
Table 6 Working time unit of each resource.

| Resource type | Max | Min | Average |
|---------------|-----|-----|---------|
| prof1         | 12  | 2   | 6       |
| prof2         | 11  | 2   | 6       |
| prof3         | 16  | 0   | 7       |
| prof4         | 80  | 76  | 76.8    |
| prof5         | 39  | 30  | 35.4    |
| prof6         | 39  | 30  | 35.4    |
| prof7         | 21  | 6   | 16.2    |
| prof8         | 15  | 1   | 4.43    |
| prof9         | 9   | 0   | 2.8     |
| prof10        | 42  | 33  | 37      |
| prof11        | 24  | 3   | 11.3    |
| prof12        | 24  | 0   | 7       |
| trans1        | 91  | 0   | 40.46   |
| trans2        | 0   | 0   | 0       |

Acknowledgments

This work was supported by Council for Science, Technology and Innovation (CSTI), Cross-ministerial Strategic Innovation Promotion Program (SIP), “Enhancement of societal resiliency against natural disasters” (Funding agency: JST).

References

[1] H. Arora, T. Raghu, and A. Vinze: Resource allocation for demand surge mitigation during disaster response, Decision Support Systems, Vol. 50, No. 1, pp. 304–315, 2010.
[2] F. Fiedrich, F. Gehbauer, and U. Rickers: Optimized resource allocation for emergency response after earthquake disasters, Safety Science, Vol. 35, No. 1-3, pp. 41–57, 2000.
[3] M. Muafa and J.E. Ramirez-Marquez: Emergency resources allocation for disaster response: An evolutionary approach, INFORMS Annual Meeting 2013, MC-1, 2013.
[4] P. Dige, C. Lund, and H.F. Ravn: Timetabling by simulated annealing, Lecture Notes in Economics and Mathematical Systems, Vol. 396, pp. 151–174, Springer, 1993.
[5] H. Kanoh, K. Hasegawa, S. Matsumoto, M. Nishihara, and N. Kato: Solving constraint satisfaction problems by a genetic algorithm adopting viral infection, IEEE International Joint Symposium on Intelligence and Systems, 1996, pp. 67–73, 1996.
[6] J.K. Lee and Y.D. Kim: Search heuristics for resource constrained project scheduling, The Journal of the Operational Research Society, Vol. 47, No. 5, pp. 678–689, 1996.
[7] K. Decker and J. Li: Coordinated hospital patient scheduling, Proceedings International Conference on Multi Agent Systems, pp. 104–111, 1998.
[8] J. Xiong, R. Leus, Z.Y. Yang, and H.A. Abbass: Evolutionary multi-objective resource allocation and scheduling in the Chinese navigation satellite system project, European Journal of Operational Research, Vol. 251, No. 2, pp. 662–675, 2015.
[9] C.F.C. Filho, D.A.R. Rocha, M.G.F. Costa, and W.C. de Albuquerque Pereira: Using constraint satisfaction problem approach to solve human resource allocation problems in cooperative health services, Expert Systems with Applications, Vol. 39, No. 1, pp. 385–394, 2012.
[10] D.E. Goldberg: Genetic Algorithms in Search, Optimization and Machine Learning, 1st ed., Addison-Wesley Longman Publishing, 1989.
[11] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi: Optimization by simulated annealing, Science, Vol. 220, pp. 671–680, 1983.
[12] A. Sadegheih: Scheduling problem using genetic algorithm, simulated annealing and the effects of parameter values on GA...
performance, *Applied Mathematical Modelling*, Vol. 30, No. 2, pp. 147–154, 2006.

[13] M.C. Hoyos, R.S. Morales, and R. Akhavan-Tabatabaei: OR models with stochastic components in disaster operations management: A literature survey, *Computers and Industrial Engineering*, Vol. 82, pp. 183–197, 2015.

[14] I. Ono and S. Kobayashi: A real coded genetic algorithm for function optimization using uni-modal normal distributed crossover, *Proceedings of the 7th International Conference on Genetic Algorithms*, pp. 246–253, 1997.

[15] H. Deguchi, M. Ichikawa, Y. Ishizuka, K. Shide, S. Someya, and Y. Yuasa: Dynamic scheduling of multi-functional resource assignment for parallel projects, *Journal of the International Association of P2M*, Vol. 6, No. 1, pp. 170–189, 2011 (in Japanese).

[16] E. Tsang: *Foundations of Constraint Satisfaction*, Academic Press, 1993.

[17] H. Sato, I. Ono, and S. Kobayashi: A new generation alternation model of genetic algorithms and its assessment, *Journal of the Japanese Society for Artificial Intelligence*, Vol. 12, No. 5, pp. 734–735, 1997.

[18] http://osm2po.de/

[19] T. Murata, H. Ishibuchi, and H. Tanaka: Multi-objective genetic algorithm and its applications to flowshop scheduling, *Computers Ind. Eng.*, Vol. 30, No. 4, pp. 957–968, 1996.

[20] I. Ono: Grid-oriented genetic algorithms for large-scale optimization, *Journal of the Society of Instrument and Control Engineers*, Vol. 47, No. 6, pp. 473–479, 2008.

[21] https://www.google.co.jp/maps/

---

**Shuang CHANG**

She received her B.Sc. degree (first class honors) and M.Phil. degree from City University of Hong Kong in 2006 and 2008 respectively, M.Sc. degree (Research) from London School of Economics and Political Science in 2010, and Ph.D. degree from Tokyo Institute of Technology in 2014. She is currently an assistant professor at Tokyo Institute of Technology. Her research interests include, but not limited to, resource allocation optimization, citizen-side disaster management, service diffusion, and public service systems design and evaluation, mainly by an agent-based approach.

**Manabu ICHIKAWA (Member)**

He received B.E., M.S., and Ph.D. degrees from Tokyo Institute of Technology in 2004, 2005, and 2009 respectively. From 2009 to 2015, he was an assistant professor at Tokyo Institute of Technology. He is currently a senior researcher at National Institute of Public Health in Japan. His research interests include, social simulation, simulation and gaming, crisis management and agent-based approach.

---

**Hiroshi DEGUCHI (Member)**

He, born in 1955 in Tokyo, is Professor in the department of Computational Intelligence and Systems Science, Tokyo Institute of Technology, Japan. He holds a Dr. of Science from Tokyo Institute of Technology and a Dr. of Economics from Kyoto University. He serves in the board of directors of several associations, e.g., Japan Association for Social and Economic Systems Studies, Pacific-Asian Association for Agent-based Approach in Social Sciences, Japan Association for Philosophy of Science, and is President of Japan Association of Simulation and Gaming (JASAG). He works as editor in chief of “Agent-Based Social Systems” (http://www.springer.com/series/7188), “Translational Systems Sciences” (http://www.springer.com/series/11213#) and is author of the book “Economics as an Agent-Based Complex System” (Springer, 2004).

**Yasuhiro KANATANI**

He, born in 1963 in Hyogo, is Director in the department of Health Crisis Management, National Institute of Public Health, Japan. He holds a Dr. of Medicine from National Defense Medical College. He serves in the member of several committees of Ministry of Health, Labour, and Welfare, Ministry of Education, Sports, Culture, Science and Technology and Ministry of Internal Affairs and Communication.