Resource Allocation and Scheduling: A Holistic Framework

Shuang Chang (Tokyo Institute of Technology) Manabu Ichikawa (National Institute of Public Health) Hiroshi Deguchi (Tokyo Institute of Technology) Yasuhiro Kanatani (National Institute of Public Health)

Abstract Resource allocation optimization and dynamic scheduling under emergent situations have been extensively yet separately studied in various fields. However, to our best knowledge, few of the extant works in these research realms provided a holistic framework to capture the allocation optimization and in-location scheduling as a series of actions from the starting place to the final destination, considering a set of constraints, such as ability-matching of divergent resources, time-distance limitations and resource scarcity. Therefore, we propose a holistic framework to integrate the resource allocation optimization and dynamic scheduling, considering the ability and capacity of different resources as well as the distance among locations in a time-efficient and supply-need matching manner. In addition, we construct an agent-based model to realize the framework by applying real-coded genetic algorithm and dynamic scheduling of multi-functional resource assignment to solve the following research problems as a holistic system: how many resources should be assigned to which location with the ability-matching property, and based on which how the limited in-location resources are scheduled.

Keyword: Resource Allocation, Dynamic Scheduling, Agent-based Approach

1 Introduction

In literature, resource allocation optimization and dynamic scheduling under emergent situations have been extensively yet separately studied in various fields such as operational research and disaster management [7, 3, 12, 6, 11, 10]. However, in some real life situations, those two aspects are closely related with each other and may require a holistic support to enable a more efficient and effective planning. To our best knowledge, few of the extant works provided a holistic framework to capture the 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. Therefore, we propose a general framework to integrate the resource allocation optimization and scheduling, considering the ability and capacity of different facilities as well as the distance in a time-efficient and ability-matching manner, and formulate this problem as a constraint satisfaction problem.

On the other side, in order to optimize resource allocation and scheduling, evolutionary algorithms, such as genetic algorithm [20] and simulated annealing [19], as well as various scheduling techniques have been extensively applied [15, 18]. In this work, in order to catch the heterogeneity of resources and to cope with the dynamic and complex process, we apply an agent-based approach by which integrating Ono's real-coded genetic algorithm [13] and Deguchi's dynamic scheduling of multi-functional resource assignment [5] to solve the following research problems: how many resources should be assigned to which location with ability-matching property, and how does the resource scheduling proceed within each location. We integrate these two research questions as one single problem such that the changing availability of resources due to the scheduling would further influence the resource allocation optimization subsequently.

This general framework indicates various potential applications. Taking disaster management as an example, this general framework could be applied to dispatch injured patients with different symptoms during a disaster to the corresponding hospital in which they could by cured while minimizing the transportation time, and subsequently based on which to schedule the transportation facilities as well as the in-hospital activities based on the capacity of resources, such as doctors, nurses and beds. Another example could be allocating the required post-disaster resources, such as food, water, and tent, to multiple temperate relief spots to minimize the gap between needs and the actual allocation.

This work is organized as follows. We model the research problem as a constraint satisfaction problem in Section 2 and propose an agent-based framework combining real-coded genetic algorithm and dynamic scheduling to solve the problems in Section 3. Some preliminary evaluation results of the framework are discussed subsequently and a conclusion with some future work will be discussed in Section 4.

2 Problem formulation

The general framework is composed by two major stages. The first stage considers how many resources should be dispatched to which location under a set of constraints to minimize a set of criteria, such as time cost and the difference between the actual allocation and resource capacity, as a constraint satisfaction problem and the following second stage is formulated as a scheduling problem considering the dynamic scheduling over time based on pre-defined activity paths and available resources. These two stages will be carried out subsequently in an iterative manner. We assume that all necessary information, such as the location of facilities, number of available resources and estimated needs, have been collected and ready to use in advance.

The detailed framework is shown in Fig. 1.

2.1 Model formulation

In this sub-section, we give a formal definition of the proposed agent-based framework. Basically we define two types of agent: agent with certain resource/ability and 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

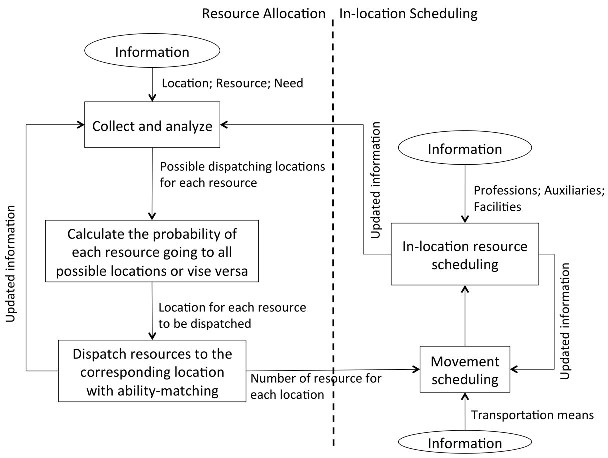


Fig. 1: A two-stage process

of disaster management again, moving emergency resources to relief spots 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.

2.1.1 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 one 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 $ability_i \in Ability = \{a_1, a_2, \ldots, a_k\}, k \in N$.

Agent with need. We also define a set of agents with certain need as $Q = \{q_1, q_2, \ldots, q_n\}$, of which each q_i possesses a need $need_i \in Need = \{ne_1, ne_2, \ldots, ne_k\}, k \in N$. This need could be resource or personnel, such as injured people needs doctors (agents with ability).

2.1.2 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 $Capacity_i^{supply} = \{c_1, c_2, \ldots, c_k \mid c_s \in N\}$, $c_s \in Capacity_i^{supply}$ is the number of agents with ability $a_s \in Ability_i$ 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 $Capacity_i^{need} = \{c_1, c_2, \ldots, c_k \mid c_s \in N\}$, $c_s \in Capacity_i^{need}$ is the number of agents with need $ne_s \in Need_i$ 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.

2.1.3 Agent moving to Spot

Both agent p_i with ability a_s and agent q_i with need ne_s possess a variable list $X_i = \{x_{ij}^s\}, \forall l_j$, s.t. $ne_s \in Need_j$ or $a_s \in Ability_j$, as the probability of going to a set of ability-matching locations, i.e. $a_s = Mapping^{-1}(ne_s)$ and $Mapping(a_s) = ne_s$. This probability could be interpreted at both micro level (as the probability of each individual) and macro level (as the proportion of agents with the same ability currently at the same location).

2.1.4 Constraint Satisfaction Problem

Depending on the situation, each agent $q_i \in Q$ with need $ne_s \in Need$ would choose a location l_j as destination where $a_s \in Ability_j$, $a_s = Mapping^{-1}(ne_s)$, or each agent p_i with ability $a_s \in Ability$ would choose a location l_j where $ne_s \in Need_j$, $ne_s = Mapping(a_s)$. 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 $D_i = \{PROB(l_1), \ldots, PROB(l_j)\}$, $\forall l_j \in L$, where $PROB : L \rightarrow [0, 1]$ is a probability function indicating the probability of choosing a location.
3. A constraint set $C = \{C_i(X_i)\}$, where $C_i(X_i) = \{l_j\}, \forall l_j \in L$ s.t. $a_s \in Ability_j$; $|C_i(X_i)|$ as the number of possible locations for this agent, and $\sum_{\forall l_t \in C_i(X_i)} PROB(l_t) = 1$.

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

2.2 Real-coded Genetic Algorithm

We apply Ono's real-coded genetic algorithm [13] to solve this constraint satisfaction problem as the first stage of the framework, aiming to minimize the time cost and the difference between expected and actual allocation while satisfying a set of ability-matching constraints. This genetic algorithm has a real number vector representation, and proposes a new crossover operation, namely unimodal normal distribution crossover (UNDX) that enables a more efficient optimization of the fitness function considering epistasis among parameters. In addition, it follows the minimal generation gap model (MGG) [17] for the population generation. The genetic algorithm representation 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 $\{l_2, l_3, l_5, l_7, l_6\}$, $\{l_2, l_4\}$, and $\{l_1, l_9, l_4\}$, respectively, then the length of the chromosome will be 10. If the agents are currently stayed in 2 locations, then the length will be 20. Each unit of the chromosome $x_{ij}^s = PROB(l_j)$ represents the probability of assigning this agent to the destination l_j satisfying the constraint $C_i(X_i)$, 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 and the difference between excepted and actual allocation. The fitness function $F(X)$ is defined as follows,

$$F(X) = min(w_1 * \sum_i \sum_j x_{ij}^s * Distance_{hj} + w_2 * \sum_i \sum_s |(c_{sj} - x_{ij}^s * N_s)|) \tag{1}$$

x_{ij}^s is agent q_i's (or p_i's) probability of going to position l_j. $Distance_{hj}$ is the actual shortest distance between location l_j (destination) and l_h (current location of agent). It could be calculated by the Dijkstra's algorithm and implemented by osm2po (an open source java package, ver. 5.0) [1]. c_{sj} is the capacity of ability a_s at location l_j, N_s is the total number of agents with need ne_s. w_1 and w_2 are weights to adjust the proportion of two objectives, whose values will be generated randomly following Murata et al. [16]. $\sum_i \sum_j x_{ij}^s * Distance_{hj}$ (as indicator 2) and $\sum_i \sum_s |(c_{sj} - x_{ij}^s * N_s)|$ (as indicator 1) are normalized as $v_i' = (v_i - v_{min})/(v_{max} - v_{min})$ before the calculation.

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

$$a_s \in Ability_j \tag{2}$$

$$\sum_j x_{ij}^s = 1, \text{for } \forall a_s \in Ability \tag{3}$$

a_s is the ability of agent p_i and $Ability_j$ is the to-be-distributed location l_j's ability set. The way to guarantee the second constraint is to set $x_{ij}^s = \frac{x_{ij}^s}{\sum_j x_{ij}^s}$ for any l_j such that $a_s \in Ability_j$.

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 large number of agents, the computation complexity of this real-coded GA design will outperform those of which the individual length depends on the number of agents, with equal number of population and similar mutation/crossover operators.

2.3 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 expected and actual allocation, based on the results of which we apply a revised version of Deguchi's dynamic scheduling of multi-functional resource assignment [5] for the actual movement and inner-location resource scheduling as a series of activities, 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 the changing situations. The resulted changes of the capacity of available resources or needs will be updated accordingly and applied to the first stage in the following cycle.

We treat each agent as a project, and the movement and in-location resource scheduling as a series of tasks. The corresponding formal definition following Deguchi [5] is given in below.

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 $\{p_1, \ldots, p_n\}$, $p_i \in P$ 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 $transpot_2$ 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 the disaster management application.

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

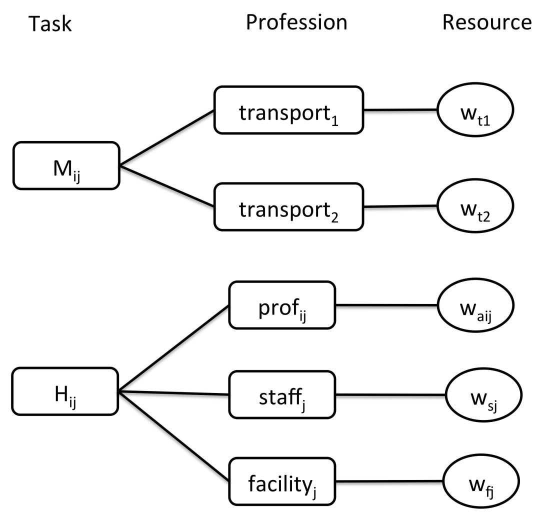


Fig. 2: Task Profession Relationship

Profession assignment function. For each profession, there is a set of available resources defined by this function, $ProfAssignF(w_{t1}) = transport_1$;$ProfAssignF(w_{t2}) = transport_2$; $ProfAssignF(w_{aij}) = prof_{ij}$, $w_{aij} \in P$ with $a_i \in Ability_j$ at location l_j; $ProfAssignF(w_{sj}) = staff_j$; $ProfAssignF(w_{fj}) = facility_j$. 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. $POSTASKS[w_{t1}] = POSTASKS[w_{t2}] = \{M_{ij}\}$; $POSTASKS[w_{aij}] = POSTASKS[w_{sj}] = POSTASKS[w_{fj}] = \{H_{ij}\}$.

Estimated Time for Tasks. $EstTimeAccompF$ is the function to assign required time unit for completing each task. $EstTimeAccompF(H_{ij})$ will be determined by assumption and $EstTimeAccompF(M_{ij})$ is

Table 1: Agents with abilities

Location	Coordination	Abilities	Capacity
l_1	35.899444; 139.623056	a,b,c,d,e	10,5,5,7,3
l_2	35.890930; 139.679232	c,d	5,10
l_3	35.809444; 139.703056	a,c,e	5,10,9
l_4	35.84325; 139.730556	a,e	8,11

Table 2: Agents with needs

Location	Coordination	Needs	Number
l_5	35.796885; 139.588938	a,b,c,d,e	36,45,78,25,41
l_6	35.838330; 139.629960	a,b,c,d,e	24,30,57,21,35
l_7	35.880363; 139.768491	a,b,c,d,e	3,5,6,3,4
l_8	35.934260; 139.719229	a,b,c,d,e	13,16,30,8,14
l_9	35.954601; 139.655366	a,b,c,d,e	2,0,2,1,1

proportional to the real distance between the departure spot and destination.

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

When assigning resources to 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 [5] 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 same amount of resources.

3 Simulation

In this section, we set up a small-scale case to experiment 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 Table 1 and Table 2. In addition, the distance between locations is estimated based on the real GIS data via Google Map [2] 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 [13]: number of crossovers for MGG: 100; population size: 500; α and β for UNDX: 0.5 and 0.35 respectively; number of generations: 2000. We run the simulation for 10 times, and the result with the least evaluation value will be passed to the dynamic scheduling phase.

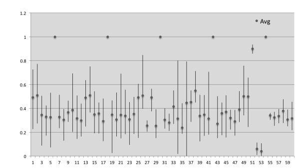
The simulation setting of the dynamic scheduling phase is as follows: each agent is treated as a project, and depending on their original location, destination

Table 3: Distance Information

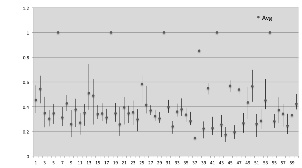
Distance (km)	l_1	l_2	l_4	l_4
l_5	13	8	18	12
l_6	16	22	14	15.5
l_7	8	14.5	8.5	11
l_8	18	24	5	15
l_9	18	11	12	7

Table 4: Dynamic Scheduling Setting

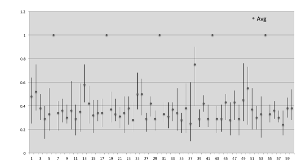
Variable	Value
$prof_{11}, prof_{13}, prof_{14}$	10,5,8
$prof_{21}$	5
$prof_{31}, prof_{32}, prof_{33}$	5,5,10
$prof_{41}, prof_{42}$	7,10
$prof_{51}, prof_{53}, prof_{54}$	3,9,11
$trans_1, trans_2$	50,5
$staff_1, staff_2, staff_3, staff_4$	30,15,24,19
$facility_1, facility_2, facility_3, facility_4$	15,10,20,5



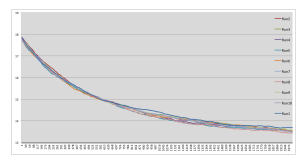
(a) Minimizing indicator 1 only



(b) Minimizing indicator 2 only



(c) Minimizing both indicators



(d) Average evaluation value of 10 runs

Fig. 3: Landscape of the probability of moving agents to each possible location

and needs, the path $path_{ijk} : M_{ij} \rightarrow H_{kj}$ of the scheduling will be determined accordingly. The value of other resources is shown in Table 4.

3.3 Simulation result

Fig. 3 (a), (b) and (c) show the landscape results of the probability of moving agents to each possible location of all 10 runs, by indicating the maximum, minimum and average value. x-axis is the 60 possible allocation cases, and y-axis is the corresponding probability. Fig. 3 (d) states 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. To calculate the number of agents, we could simply multiple the probability with the number of agents with each need.

Fig. 4 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 expected and actual allocation, x-axis is the evaluation value composing both indicators; 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. Fig. 4 (a), (b), (c) show respectively the situation that only indicator 1, indicator 2 and both indicators are optimized. The x-axis and y-axis represent the evaluation value of indicator 1 and 2 respectively in Fig. 4 (d), 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. 5 (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. 5 (b)). By analyzing the tables, we could identify

Table 5: Probability of agents moving to the corresponding spot

Location	Needs	Possible destination	Probability
	a	l_1,l_3,l_4	0.44,0.15,0.42
	b	l_1	1
l_5	c	l_1,l_2,l_3	0.37,0.35,0.27
	d	l_1,l_2	0.58,0.42
	e	l_1,l_3,l_4	0.40,0.31,0.28
	a	l_1,l_3,l_4	0.38,0.31,0.32
	b	l_1	1
l_6	c	l_1,l_2,l_3	0.35,0.37,0.28
	d	l_1,l_2	0.63,0.37
	e	l_1,l_3,l_4	0.43,0.27,0.29
	a	l_1,l_3,l_4	0.42,0.28,0.3
	b	l_1	1
l_7	c	l_1,l_2,l_3	0.33,0.3,0.37
	d	l_1,l_2	0.59,0.41
	e	l_1,l_3,l_4	0.32,0.32,0.36
	a	l_1,l_3,l_4	0.29,0.48,0.24
	b	l_1	1
l_8	c	l_1,l_2,l_3	0.23,0.27,0.5
	d	l_1,l_2	0.15,0.85
	e	l_1,l_3,l_4	0.33,0.45,0.22
	a	l_1,l_3,l_4	0.21,0.44,0.35
	b	l_1	1
l_9	c	l_1,l_2,l_3	0.31,0.34,0.34
	d	l_1,l_2	0.49,0.51
	e	l_1,l_3,l_4	0.26,0.4,0.35

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.

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.

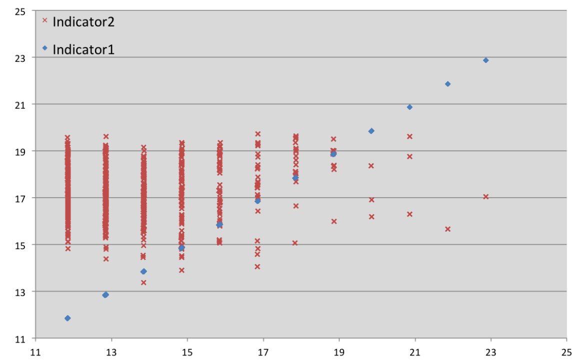
4 Conclusion

This work proposed a general holistic framework for integrating ability-matching resource allocation and scheduling under the time and resource limitations, and constructed an agent-based model applying real-coded GA and dynamic scheduling to optimize the raised problems. In addition, we evaluated the framework by setting up a small scale case, through which we could have a very straight-forward optimized allocation plan and the scheduling of each resource as a dynamic process, and trace the results of any change due to emerging or unexpected situations. This preliminary holistic framework is expected to contribute significantly to the emergency resource allocation and scheduling to resolve problems stemmed from real life applications.

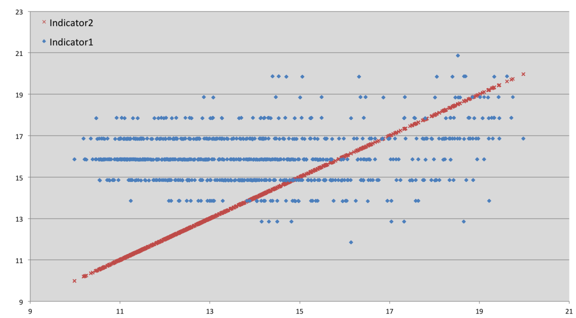
In future work, with any available empirical data, we could further improve the framework and apply it to real situations to capture the dynamic process and emerging situations in real-time. By experimenting more scenarios, the insight and knowledge gained could facilitate the training of rescue teams and the policy-making process of multiple involved stakeholders.

5 Acknowledgement

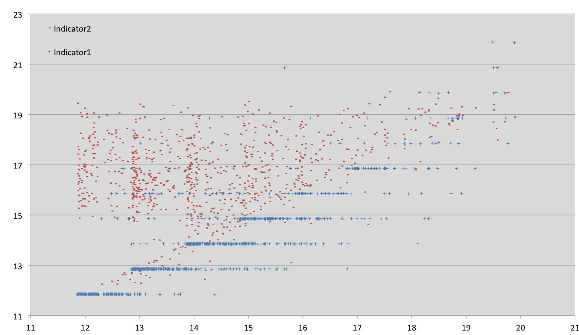
This work was supported by Council for Science, Technology and Innovation(CSTI), Cross-ministerial Strategic Innovation Promotion Program (SIP), "Enhancement of societal resiliency against natural disasters" (Funding agency:JST).



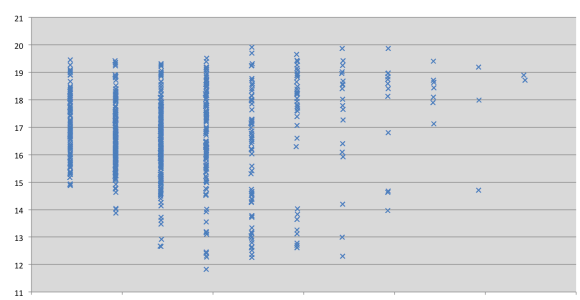
(a) Considering only indicator 1



(b) Considering only indicator 2



(c) Considering both indicators



(d) The relation between indicator 1 and indicator 2

Fig. 4: The evaluation value of both indicators

References

1) http://osm2po.de/

2) https://www.google.co.jp/maps/

3) H. Arora, T. Raghu, and A. Vinze: Resource allocation for demand surge mitigation during disaster response. Decision Support Systems, 50(1), 304/315 (2010)

#DtalgebraTable2										
PT#	ProjectName	TaskPathNar	AllowedStart	TaskName	WorkNum	ConstrainProf	ResourceCoc	ResourceNar	Profession	StartTime
decimal	string	string	decimal	string	decimal	string	string	string	string	decimal
#	#	#	#	#	#	#	#	#	#	#
#	#	#	#	#	#	#	#	#	#	#
1	p1	path111	0	M11	4	trans11	t1010	t1010	trans11	15
2	p1	path111	0	H11	2	prof11,staff1,Bed1	w1002	w1002	prof11	19
2	p1	path111	0	H11	2	prof11,staff1,Bed1	s1002	s1002	staff1	19
2	p1	path111	0	H11	2	prof11,staff1,Bed1	b1002	b1002	Bed1	19
3	p2	path111	0	M11	4	trans11	t1010	t1010	trans11	19
4	p2	path111	0	H11	2	prof11,staff1,Bed1	w1001	w1001	prof11	23
4	p2	path111	0	H11	2	prof11,staff1,Bed1	s1001	s1001	staff1	23
4	p2	path111	0	H11	2	prof11,staff1,Bed1	b1001	b1001	Bed1	23
5	p3	path111	0	M11	4	trans11	t1001	t1001	trans11	20
6	p3	path111	0	H11	2	prof11,staff1,Bed1	w1002	w1002	prof11	24
6	p3	path111	0	H11	2	prof11,staff1,Bed1	s1002	s1002	staff1	24
6	p3	path111	0	H11	2	prof11,staff1,Bed1	b1002	b1002	Bed1	24
7	p4	path111	0	M11	4	trans11	t1002	t1002	trans11	20
8	p4	path111	0	H11	2	prof11,staff1,Bed1	w1003	w1003	prof11	24
8	p4	path111	0	H11	2	prof11,staff1,Bed1	s1003	s1003	staff1	24
8	p4	path111	0	H11	2	prof11,staff1,Bed1	b1003	b1003	Bed1	24
9	p5	path111	0	M11	4	trans11	t1003	t1003	trans11	20
10	p5	path111	0	H11	2	prof11,staff1,Bed1	w1004	w1004	prof11	24
10	p5	path111	0	H11	2	prof11,staff1,Bed1	s1004	s1004	staff1	24
10	p5	path111	0	H11	2	prof11,staff1,Bed1	b1004	b1004	Bed1	24
11	p6	path111	0	M11	4	trans11	t1004	t1004	trans11	20
12	p6	path111	0	H11	2	prof11,staff1,Bed1	w1005	w1005	prof11	24
12	p6	path111	0	H11	2	prof11,staff1,Bed1	s1005	s1005	staff1	24
12	p6	path111	0	H11	2	prof11,staff1,Bed1	b1005	b1005	Bed1	24
13	p7	path111	0	M11	4	trans11	t1005	t1005	trans11	20

(a) Assigned resources and the start time of each project

#DtalgebraTable2										
PT#	ProjectName	TaskName	0	1	2	3	4	5	6	7
decimal	string	string	string	string	string	string	string	string	string	string
#	#	#	#	#	#	#	#	#	#	#
#	#	#	#	#	#	#	#	#	#	#
32	p16	H11								
32	p16	H11								
33	p17	M13	M13[p17]/t1001	M13[p17]/t1001	M13[p17]/t1001	M13[p17]/t1001	M13[p17]/t1001			
34	p17	H13						H13[p17]/w3001	H13[p17]/w3001	
34	p17	H13						H13[p17]/s3001	H13[p17]/s3001	
34	p17	H13						H13[p17]/b3001	H13[p17]/b3001	
35	p18	M13	M13[p18]/t1002	M13[p18]/t1002	M13[p18]/t1002	M13[p18]/t1002	M13[p18]/t1002			
36	p18	H13						H13[p18]/w3003	H13[p18]/w3003	
36	p18	H13						H13[p18]/s3004	H13[p18]/s3004	
36	p18	H13						H13[p18]/b3004	H13[p18]/b3004	
37	p19	M13	M13[p19]/t1003	M13[p19]/t1003	M13[p19]/t1003	M13[p19]/t1003	M13[p19]/t1003			
38	p19	H13						H13[p19]/w3004	H13[p19]/w3004	
38	p19	H13						H13[p19]/s3005	H13[p19]/s3005	
38	p19	H13						H13[p19]/b3005	H13[p19]/b3005	
39	p20	M13	M13[p20]/t1004	M13[p20]/t1004	M13[p20]/t1004	M13[p20]/t1004	M13[p20]/t1004			
40	p20	H13						H13[p20]/w3005	H13[p20]/w3005	
40	p20	H13						H13[p20]/s3006	H13[p20]/s3006	
40	p20	H13						H13[p20]/b3006	H13[p20]/b3006	
41	p21	M13	M13[p21]/t1005	M13[p21]/t1005	M13[p21]/t1005	M13[p21]/t1005	M13[p21]/t1005			
42	p21	H13							H13[p21]/w3002	H13[p21]/w3002
42	p21	H13							H13[p21]/s3002	H13[p21]/s3002
42	p21	H13							H13[p21]/b3002	H13[p21]/b3002
43	p22	M14								
44	p22	H14								

(b) Assigned resources and time units of each project

#DtalgebraTable2															
ResourceCoc	ResourceName	PT#	ProjectName	0	1	2	3	4	5	6	7	8	9	10	11
string	string	decimal	string	string	string	string	string	string	string	string	string	string	string	string	string
#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
b2003	b2003	264	p132												
b2003	b2003	274	p137												
b2003	b2003	606	p303								H32[p303]	H32[p303]	H32[p303]		
b2003	b2003	616	p308											H32[p308]	H32[p308]
b2003	b2003	626	p313												
b2003	b2003	636	p318												
b2003	b2003	804	p402					H32[p402]	H32[p402]	H32[p402]					
b2003	b2003	902	p451												
b2003	b2003	956	p478												
b2004	b2004	228	p114												
b2004	b2004	236	p118												
b2004	b2004	246	p123												
b2004	b2004	256	p128												
b2004	b2004	266	p133												
b2004	b2004	598	p299						H32[p299]	H32[p299]	H32[p299]				
b2004	b2004	608	p304									H32[p304]	H32[p304]	H32[p304]	
b2004	b2004	618	p309												H32[p309]
b2004	b2004	628	p314												
b2004	b2004	638	p319												
b2004	b2004	814	p407					H42[p407]							
b2004	b2004	904	p452												
b2005	b2005	238	p119												
b2005	b2005	248	p124												
b2005	b2005	258	p129												
b2005	b2005	268	p134												
b2005	b2005	600	p300						H32[p300]	H32[p300]	H32[p300]				
b2005	b2005	610	p305									H32[p305]	H32[p305]	H32[p305]	
b2005	b2005	620	p310												H32[p310]

(c) Assigned projects and occupied time units of each resource

Fig. 5: Resulted tables of the dynamic scheduling (part)

4) K. Decker and J. Li: Coordinated hospital patient scheduling. In Multi Agent Systems, 1998. Proceedings. International Conference on, 104/111 (1998)

5) 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, 6(1), 170/189 (2011)

6) P. Dige, C. Lund, and H. F. Ravn: Timetabling by simulated annealing. Lecture Notes in Economics and Mathematical Systems, vol(396), 151/174 (1993)

7) F. Fiedrich, F. Gehbauer, and U. Rickers: Optimized resource allocation for emergency response after earthquake disasters. Safety Science, 35(1-3), 41/57 (2000)

8) C. F. 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, 39(1), 385/394 (2012)

9) 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, 82, 183/197 (2015)

10) H. Kanoh, K. Hasegawa, S. Matsumoto, M. Nishihara, and N. Kato: Solving constraint satisfaction problems by a genetic algorithm adopting viral infection. In Intelligence and Systems, 1996., IEEE International Joint Symposia, 67/73 (1996)

11) J. K. Lee and Y. D. Kim: Search heuristics for resource constrained project scheduling. The Journal of the Operational Research Society, 47(5) (1996)

12) M. Muaafa and J. E. Ramirez-Marquez: Emergency resources allocation for disaster response: An evolutionary approach. In INFORMS Annual Meeting 2013, Minneapolis, MN (2013)

13) I. Ono and S. Kobayashi: A real coded genetic algorithm for function optimization using unimodal normal distributed crossover. In Proceedings of the 7th International Conference on Genetic Algorithms, East Lansing, MI, USA, July 19-23, 246/253 (1997)

14) E. Tsang: Foundations of Constraint Satisfaction. Academic Press (1993)

15) 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. In European Journal of Operational Research (2015)

16) T. Murata, H. Ishibuchi, and H. Tanaka: Multi-objective genetic algorithm and its applications to flowshop scheduling. Computers ind. Engng, vol. 30(4), 957/968 (1996)

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, 12(5), 734/735 (1997)

18) A. Sadegheih: Scheduling problem using genetic algorithm, simulated annealing and the effects of parameter values on GA performance. Applied Mathematical Modelling, 30(2), 147/154 (2006)

19) S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi: Optimization by simulated annealing. Science, vol. 220, 671/680 (1983)

20) David E. Goldberg: Genetic Algorithms in Search, Optimization and Machine Learning (1st ed.). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1989)