An Efficient Outpatient Scheduling Approach

Haibin Zhu, *Senior Member, IEEE*, Ming Hou, *Senior Member, IEEE*,
Chun Wang, *Member, IEEE*, and MengChu Zhou, *Fellow, IEEE*

***Abstract*—Outpatient scheduling is considered as a complex problem. Efficient solutions to this problem are required by many health care facilities. This paper proposes an efficient approach to outpatient scheduling by specifying a bidding method and converting it to a group role assignment problem. The proposed approach is validated by conducting simulations and experiments with randomly generated patient requests for available time slots. The major contribution of this paper is an efficient outpatient scheduling approach making automatic outpatient scheduling practical. The exciting result is due to the consideration of outpatient scheduling as a collaborative activity and the creation of a qualification matrix in order to apply the group role assignment algorithm.**

***Note to Practitioners*—As the "Age Wave" approaches, health care facilities are becoming relatively scarce worldwide compared with what are demanded. The varying availability, requirements, and preferences of both facilities and outpatients make the problem of scheduling outpatient appointments on health care facilities extremely challenging. Traditional manually operated scheduling systems based on phone calls are out of date although they are still widely used due to lack of new effective scheduling systems. To solve such a problem requires an efficient Web-based system to schedule the appointments instantly. The proposed approach provides a technical foundation for efficient Web-based scheduling systems, which can be applied directly to not only outpatient scheduling in the health care sector, but also in some other real-world scheduling applications.**

***Index Terms*—Agents, outpatient scheduling, role assignment, roles.**

Manuscript received March 22, 2012; revised June 07, 2012; accepted June 18, 2012. Date of publication August 15, 2012; date of current version October 02, 2012. This paper was recommended for publication by Associate Editor C. Chu and Editor H. Ding upon evaluation of the reviewers' comments. This work was supported in part by NSERC under Grant 262075-2012, IBM Eclipse Innovation Grant, and DRDC-Toronto Grant.

H. Zhu is with the Department of Computer Science and Mathematics, Nipissing University, North Bay, ON P1B 8L7, Canada (e-mail: haibinz@nipissingu.ca).

M. Hou is with Defence Research and Development Canada-Toronto, Toronto, ON M3M 3B9, Canada (e-mail: Ming.Hou@drdc-rddc.gc.ca).

C. Wang is with the Concordia Institute for Information Systems Engineering, Concordia University, Montreal, H3G 1M8, Canada (e-mail: cwang@ciise.concordia.ca).

M. C. Zhou is with the MoE Key Laboratory of Embedded System and Service Computing, Tongji University, Shanghai 200092, China (e-mail: mengchu@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors. This includes a powerpoint file, which shows the simulation, the performance comparison, the performance verification, a sample simulation output, the ILOG CPL models and data, the experiment results, the graphs of time trends, and the distribution of times of random cases. This material is 2 MB in size.

Digital Object Identifier 10.1109/TASE.2012.2207453

I. Introduction

In HEALTH CARE management, some facilities, such as magnetic resonance imaging (MRI) scanning or computed tomography (CT) scanning, are expensive and critical for certain disease diagnoses [2], [19]. Normally, expensive facilities must be prescheduled for unanticipated requirements of inpatients. Sometime later, prescheduled time slots may become available for outpatients to use. At the same time, outpatients may have different requests for the appointments and some appointments may not be available for some time slots, i.e., there are different constraints in assigning available time slots of an expensive device to available patients.

Therefore, instant rescheduling is required at such a moment in terms of volume and urgency. Because the problem of efficient scheduling of patient appointments on expensive resources is complex and dynamic, it must be solved with an efficient system to reschedule the appointments to avoid wastes and make full use of the expensive and critical facilities, i.e., the objective of outpatient scheduling is to find an appointment system for which a particular measure of performance is optimized in a clinical environment—an application of resource scheduling under uncertainty [2].

In role-based collaboration (RBC) [23], group role assignment (GRA) [27] is a complex problem for which the exhaustive-search based algorithm has exponential complexity. An efficient algorithm for GRA has been developed based on the Hungarian algorithm, also called Kuhn-Munkres (K-M) algorithm [11], [14]. It is of polynomial complexity. This work builds a system that transfers outpatient scheduling into a GRA problem.

II. Related Work

It is well accepted that scheduling problems in health care services are important and complex. Much research is conducted in the fields of operational research and industrial engineering [2], [5]. However, very few attempts have been made to solve them through the development of practical systems. Cayirli and Veral [2] present general problem formulation and modeling considerations for effective scheduling systems, and provide taxonomy of methodologies used in previous literature. Godin and Wang [5] propose to allocate available diagnostic services timeslots to outpatients through an iterative bidding procedure which is a trigger to the idea of call-for-collaboration (CFC) in this paper. Gul *et al.* [6] compare several heuristics for scheduling an outpatient procedure center with respect to the competing criteria of expected patient waiting time and overtime. Guo *et al.* [7] present a simulation framework for the evaluation and optimization of scheduling rules. Gupta and Denton [8] state that many factors affect the performance of such systems including arrival

TABLE I
AN EXAMPLE OF A TWO-WEEK SCHEDULE

	Mon	Tue	Wed	Thu	Fri	Mon	Tue	Wed	Thu	Fri
8:00	**0**	0	0	0	0	0	0	0	0	0
9:00	**1**	1	**1**	1	1	1	1	**1**	1	1
10:00	2	**2**	**2**	2	2	2	2	**2**	2	2
11:00	3	**3**	**3**	3	3	3	3	**3**	**3**	3
12:00	4	**4**	4	4	4	4	4	4	**4**	4
13:00	5	5	5	5	5	5	5	**5**	5	5
14:00	6	6	6	**6**	6	6	**6**	**6**	**6**	6
15:00	**7**	7	7	**7**	7	7	**7**	**7**	**7**	7
16:00	**8**	8	8	**8**	8	8	**8**	**8**	**8**	8
17:00	**9**	9	9	9	9	9	**9**	9	**9**	9
18:00	**10**	10	10	10	10	10	10	10	**10**	10
19:00	11	11	11	11	11	11	11	11	**11**	11

and service time variability, patient and provider preferences, available information technology and the experience level of the scheduling staff. Kaandorp and Koole [10] model the outpatient appointment scheduling problem, and present a method to obtain optimal outpatient schedules in case of a finite number of possible arrivals. They do not mention the efficiency of their algorithms. Liu *et al.* [13] develop a framework and propose heuristic dynamic policies for scheduling patient appointments by considering that patients may cancel or not show up for their appointments. Patrick and Puterman [16] analyze the CT operations at Vancouver General Hospital, Canada and state that inpatient demands are fluctuating and the outpatients have different priorities [2]. Our example is established based on this work [16]. Santibáñez *et al.* [19] develop a mixed integer programming model to schedule surgical blocks for each specialty into operating rooms and applied it to the hospitals in British Columbia Health Authority, considering operating room time availability and postsurgical resource constraints. Vermeulen *et al.* [20] present an adaptive approach to automatic optimization of resource calendars. The allocation of capacity to different patient groups is flexible and adaptive to the current and expected future situation.

III. A REAL-WORLD OUTPATIENT SCHEDULING PROBLEM

The MRI lab of a hospital has a future two-week schedule from Monday to Friday. Because some preoccupied slots are released by inpatients, the newly available hours (slots) are presented as the shaded cells in Table I. Note that the preoccupied slots cannot be freed earlier than a period of time, e.g., two weeks, because the slots must be prepared for the inpatients. Hence, a two-week schedule is usually adopted.

Now, only one work day is left before the new schedule. The administration hopes to use as many unoccupied slots as possible. The question is how to assign the available slots to the most needed outpatients and instantly informs the outpatients the new scheduled time slots.

Conventionally, outpatient records in a database tell some outpatients' prefilled requirements and availability in choosing adjacent time slots (also called requested bundles of time slots, or simply, requested bundles). Those outpatients who can come to fill newly available slots in Table I are shown in Table III. Based on such a table, an outpatient scheduling problem is modeled as an optimization problem and proved as an NP-hard problem [5]. To formalize the questions in this paper, we use the symbols in Table II.

TABLE II
SYMBOLS USED IN THIS PAPER

Symbol	Meaning
$\mathcal{A}$	A set of agents.
B	A bundle of time slots.
$F^t(C, j)$	The set formed by the elements of vector $C[j]$.
$\mathcal{N}$	The set of non-negative integers, i.e., {0, 1, 2, …}.
P	An n-dimensional vector of vectors of the bidding blocks from outpatients. P_j is a vector with k_j ($1 \le k_j \le m$) bidding blocks requested by patient $j \in \Pi$. P_j also expresses the preferences of patient j (if $l < k$ then patient j prefers the l^{th} block $P_j[l]$ to the k^{th} one $P_j[k]$, $0 \le l, k < k_j, l \ne k$).
P'	An n-dimensional vector of vectors of bidding sets (also called bundles) of time slots from outpatients, where P'_j is a vector with k_j ($1 \le k_j \le m'$) bidding sets (bundles) of time slots requested by patient $j \in \Pi$.
Q: $\mathcal{A} \times \mathcal{R} \to [0, 1]$	A qualification matrix. $Q[i, j]$ expresses the qualification value of agent I for role j.
$\mathcal{R}$	A set of roles.
S: $\Omega \to \mathcal{N}$	$S[i]$ expresses the size of block I ($I \in \Omega$).
S': $\Pi \to \mathcal{N}$	$S'[j]$ expresses the requested block size by patient j ($j \in \Pi$).
T: $\mathcal{A} \times \mathcal{R} \to \{0, 1\}$	An assignment matrix. $T[I, j] = 1$ means that agent I is assigned to role j, and $T[I, j] = 0$ otherwise.
T'	A vector of appointments for outpatients.
V: $\mathcal{R} \to \mathcal{A}$	$V[j]$ expresses the original agent assigned to role j.
W: $\mathcal{R} \to \mathcal{N}$	A vector to express the priority values of outpatients (roles), where W_j be the priority value assigned to patient $j \in \Pi$ or $j \in \mathcal{R}$.
g: $\Omega \times \Pi \to [0, 1]$	A preference scale to evaluate the relative preference among different patients and requested blocks.
H: $\Omega \times \Pi \to [0, 1]$	The fitness of the requested size of the requested block from j versus the available size of block i.
I	An element in Ω or $\mathcal{A}$.
J	An element in Π or $\mathcal{R}$.
K	$max\{m, n\}$
m'	The number of elements in Ω', i.e., $\lvert\Omega'\rvert$.
m	The number of elements in Ω or $\mathcal{A}$, i.e., $\lvert\Omega\rvert$ or $\lvert\mathcal{A}\rvert$.
n	The number of elements in Π or $\mathcal{R}$, i.e., $\lvert\Pi\rvert$ or $\lvert\mathcal{R}\rvert$.
w_{max}	The maximum number in W.
X	An assignment indicator. $X_j(B)=1$ if bundle B is allocated to patient j ($B \subseteq \Omega'$, $0 \le j < n$), $x_j(B)=0$ otherwise.
Δ	The sum of the priority values of the scheduled patients, i.e., $\sum_{B \in P'} \sum_{j \in \Pi} x_j(B) \times W_j$.
Δ_1	The sum of the priority values of on the scheduled time slots, i.e., $\sum_{i \in \Omega} \sum_{j \in \Pi} x_j(i) \times S'[j] \times W_j$.
Δ_1^*	max $\{\Delta_1\}$.
Π	A set of the bidding outpatients.
Ω	A set of available time slot blocks, or simply blocks.
Ω', Ω''	A set of available time slots.
δ	$min\{m, n\}$
$\zeta(P, i, j)$	To express if agent i belongs to $F^t(C, j)$.
Ψ	$\sum_{j \in \Pi} S'[j] \times W_j$.

TABLE III
AVAILABILITIES AND PREFERENCES OF OUTPATIENTS

ID	Name	Priority value	Outpatients' availabilities in a day
P1	Tom	3	Monday: {7, 8}, Wednesday: {2, 3}, Thursday: {3, 4}
P2	Chris	2	Monday and Tuesday: {7,8,9}
P3	Ana	3	Wednesday and Thursday: {3}
P4	Bob	2	Wednesday: {1,2}, Thursday: {3, 4}
P5	Don	1	Any day: {0},…, {11}
P6	Jane	3	Thursday: {7, 8}

TABLE IV
AVAILABLE TIME SLOTS (34)

	Mon	Tue	Wed	Thu	Fri	Mon	Tue	Wed	Thu	Fri
8:00	0									
9:00	1		9					19		
10:00		6	10					20		
11:00		7	11					21	26	
12:00		8							27	
13:00								22		
14:00				12			15	23	28	
15:00	2			13			16	24	29	
16:00	3			14			17	25	30	
17:00	4						18		31	
18:00	5								32	
19:00									33	

TABLE V
AVAILABILITIES AND PREFERENCES OF OUTPATIENTS

Patient ID	Name	Priority value	Outpatients' requested bundles
P1	Tom	3	{2, 3},{9, 10},{19, 20}, {26, 27}
P2	Chris	2	{2, 3, 4},{16, 17, 18}
P3	Ana	3	{11}, {21}, {26}
P4	Bob	2	{9, 10},{19, 20}, {26,27}
P5	Don	1	{0},…, {33}
P6	Jane	3	{13, 14}, {29, 30}

TABLE VI
ASSIGNED TIME SLOTS (NAMED SLOTS)

	Mon	Tue	Wed	Thu	Fri	Mon	Tue	Wed	Thu	Fri
8:00	Don									
9:00	1		Bob					19		
10:00		6	Bob					20		
11:00		7	Ann					21	26	
12:00		8							27	
13:00								22		
14:00				12			15	23	28	
15:00	Tom			Jane			Chris	24	29	
16:00	Tom			Jane			Chris	25	30	
17:00	4						Chris		31	
18:00	5								32	
19:00									33	

If each outpatient requires one bundle of time slots and we do not consider the preferences of patients among requested bundles, the problem can be formalized as [5]

$$\max \Delta = \sum_{B\in P'} \sum_{j\in\Pi} x_j(B) \times W_j \tag{1}$$

subject to

$$\forall j \in \Pi, \sum_{B\subseteq\Omega'} x_j(B) \le 1 \tag{2}$$

$$\forall i \in \Omega', \sum_{B\ni i} \sum_{j\in\Pi} x_j(B) \le 1 \tag{3}$$

$$\forall j \in \Pi, \sum_{B\subseteq\Omega'} x_j(B) = \sum_{B\in P'_j} x_j(B) \tag{4}$$

$$\forall B \subseteq \Omega', j \in \Pi x_j(B) \in \{0, 1\} \tag{5}$$

where constraint (2) ensures that each outpatient can be assigned at most one requested bundle of timeslots; (3) ensures that each time slot is assigned to only one patient; (4) ensures that the assigned bundle is requested by the patient; and (5) is a 0–1 constraint. Note that in $\sum_{B\ni i}$ (3) is read as "sum for all B that contains i."

To solve the problem described in (1)–(5), the available time slots are numbered from 0–33 (Table IV), outpatients are numbered from 0–5, and their availability is transferred to Table V.

By using IBM ILOG CPLEX Optimization Studio V12.2 (ILOG), the problem is solved (Table VI) in 130 ms with the objective as 26 (Section V of the multimedia document). Experiments assert that such modeling could only work for relatively small size outpatient scheduling problems [5].

IV. COLLABORATIVE OUTPATIENT SCHEDULING—OUR STRATEGY

In fact, outpatient scheduling is dynamic. Available time slots, the outpatients' availability and preference are changing. The information in Table III may not reflect the current states of all the outpatients. The model in (1)–(5) does not consider the preferences of outpatients.

In our strategy, the outpatient scheduling problem is considered as a collaborative action, i.e., the patients are collaborating on this scheduling work. The operation scenario is that, upon a change in the available time slots, the facility office or clinic sends out a CFC message by e-mails or calls to all the already registered (scheduled or not yet scheduled) outpatients, and some or all of them respond to the CFC message by bidding for bundles of time slots. The scheduling algorithm then makes optimal rescheduling based on their responses.

We assume that the time unit is in slot, and each outpatient has a priority value assigned by his/her doctor. The priority values of outpatients are taken as one indicator to express the ranking and competence on a time slot among outpatients. In this way, if a bundle of two time slots is assigned to a patient who has a priority value of 3, we collect 3 × 2 into *the sum of the priority values on the assigned time slots*.

Then, the outpatient scheduling system is designed to *find an assignment scheme for all patients such that the sum of priority values on the assigned slots is maximized and their preferences are satisfied*.

We assume that the bidding patients may have their original appointments when CFC is initiated, and they will contribute available slots when they are rescheduled. CFC is a process that is initiated manually or automatically according to a schedule or

TABLE VII
THE AVAILABLE TIME BLOCKS (10) FOR TABLE I

	Mon	Tue	Wed	Thu	Fri	Mon	Tue	Wed	Thu	Fri
8:00	0									
9:00			3					6		
10:00		2								
11:00									8	
12:00										
13:00								7		
14:00				4			5		9	
15:00	1									
16:00										
17:00										
18:00										
19:00										

newly available information, such as, time slots become available and it is needed to reschedule. It ends when the iteration of rescheduling is done. The system process is described as follows, where $\Omega''(\subseteq \Omega')$ is a new set of available time slots after one iteration of rescheduling, $\Omega' = \Omega''$ expresses that no new slot is allocated in the rescheduling process.

CallForCollaboration Process:

Input: Ω'

Output: T'

Repeat

Step 1: Receive: Π, W and P';

Until (time is due).

Repeat

Step 2: *Rescheduling* $(\Omega', \Pi, W, P', \Omega'', T')$;

Until $(\Pi = \Phi \vee \Omega' = \Omega'')$;

Send out or post new schedules T';

Rescheduling Process:

Input: Ω', Π, W, P'

Output: T' and Ω''.

Step 1: Maximize Δ while their preferences are satisfied;

Step 2: Form and return T' and Ω''.

To decrease the search space in the rescheduling process, we propose to introduce some restrictions in a bidding process.

1) Each round of CFC considers a group of continuous time slots as a whole, called time slot block or simply block with a size attached (i.e., the number of continuous time slots). Now, we obtain a set of available blocks Ω. A size vector $S : \Omega \rightarrow \mathcal{N} = \{0, 1, 2, \ldots\}$ is introduced, where $S[i]$ expresses the size of block $i (i \in \Omega)$.
2) Patients may choose some from these blocks (see Table VII, the available blocks have different number of slots) and specify how many slots they require. A size vector, $S' : \Pi \rightarrow \mathcal{N}$ is introduced, where $S'[j]$ expresses the requested size of the requested block by patient $j (j \in \Pi)$.
3) The sequence of the outpatients' choices shows their preferences.
4) A patient is allocated at most one block with the requested size.

As for the problem in Table I, we may redraw it shown as in Table VII. Now, the problem can be respecified. We introduce a function $g : \Omega^* \times \Pi \rightarrow [0, 1]$, where Ω^* is the power set of Ω, as a preference scale to evaluate the relative preference among different patients and requested bundles, i.e., $g(i, j) =$

$$\begin{cases} \frac{(k_j - l)}{k_j}, & l = 0, 1, \ldots, k_j - 1, \ i \in P_j, \ i = P_j[l]; \\ 0, & \text{otherwise} \end{cases} . \tag{6}$$

$h(i, j)$ is introduced to evaluate the matching scales of the request, i.e.,

$$h(i, j) = \begin{cases} \frac{S'[j]}{S[i]} & (S'[j] \le S[i]) \\ 0 & (S'[j] > S[i]) \end{cases} \quad (0 \le i < \mathrm{m}, 0 \le j < \mathrm{n}) \tag{7}$$

For example, suppose that other conditions are the same. If patient x bids for the third $(i = 2)$ choice in five choices $(k_x = 5)$; and patient y bids for the second $(i = 1)$ in three choices $(k_y = 3)$, we prefer the latter, i.e., y, because $g(2, x) = 3/5 < g(2, y) = 2/3$. If patient x requests three slots in a five-slot block z, i.e., $S[z] = 5$, $S'[x] = 3$, and patient y requests four slots in the same block z, $S'[y] = 4$, we prefer y, because $h(z, x) = S'[x]/S[z] = 3/5 < h(z, y) = S'[y]/S[z] =4/5$.

With this adjustment, the outpatient scheduling problem becomes *to find an assignment scheme for all patients such that Δ_1 is maximized; all patient preferences are best satisfied; and all patient requests are best matched*.

It can be reformalized as a three-objective optimization problem

$$\max \Delta_1 = \sum_{i \in \Omega} \sum_{j \in \Pi} x_j(i) \times S'[j] \times W_j \tag{8}$$

$$\max \sum_{i \in \Omega} \sum_{j \in \Pi} x_j(i) \times g(i, j) \tag{9}$$

$$\max \sum_{i \in \Omega} \sum_{j \in \Pi} x_j(i) \times h(i, j) \tag{10}$$

subject to

$$\forall j \in \Pi, \sum_{i \in \Omega} x_j(i) \le 1 \tag{11}$$

$$\forall j \in \Pi, \sum_{i \in \Omega} x_j(i) = \sum_{i \in P_j} x_j(i) \tag{12}$$

$$\forall i \in \Omega, \sum_{j \in \prod} x_j(i) \times S'[j] \le S[i] \tag{13}$$

$$\forall i \in \Omega, j \in \Pi, x_j(i) \in \{0, 1\} \tag{14}$$

where (11) ensures that any outpatient can only obtain one from the available time blocks; (12) ensures that if a timeslot block is assigned to an outpatient, it must belong to the block set the outpatient has requested; constraint (13) tells that each block can be allocated to more than one patient, however, the total requested sizes of the assigned patients should not be larger than the size of the original block; and (14) is a 0–1 constraint. It is evident that the bundle requirement is removed and therefore the problem is simplified.

Note that this is a typical multiobjective optimization problem to which a simple solution is weighted sum [16]. However, it is still time-consuming based on the model (6)–(14).

TABLE VIII
AN EXAMPLE FOR OUTPATIENT SCHEDULING

ID	Name	Priority value	Patients' biddings for blocks	Block size	Original appointment
P1	Tom	3	1, 3, 6, 8	2	14 (8-9 of Thursday in the 3rd week)
P2	Chris	2	1, 5	3	12 (12-14 of Monday in the 4th week)
P3	Ana	3	3, 6, 8	1	10 (10 of Thursday in the 3rd week)
P4	Bob	2	3, 6, 8	2	13(10-11 of Tuesday in the 4th week)
P5	Don	1	0,…, 9	1	11(17 of Friday in the 3rd week)
P6	Jane	3	4, 9	2	15 (13-14 of Friday in the 4th week)

TABLE IX
ASSIGNED TIME SLOTS

	Mon	Tue	Wed	Thu	Fri	Mon	Tue	Wed	Thu	Fri
8:00	Don									
9:00			Tom					Bob		
10:00			Tom					Bob		
11:00									Ann	
12:00										
13:00										
14:00				Jane						
15:00	Chris			Jane						
16:00	Chris									
17:00	Chris									
18:00										
19:00										

For example, a random case ($m, n = 14$) cost 54 minutes to be solved by a weighted sum method by using ILOG (The Model and Data I in Section VI of the multimedia document).

Table VIII is the assumed information collected by one round of CFC similar to the requests from Table V. It is assumed that the responses to CFC are for the available blocks from 0 to 9. The blocks originally assigned to the 6 responded patients are 10–15. *Please note that it is not hard to collect the information shown in* Table VIII. For example, a group e-mail can be sent out to all the outpatients. The interested ones can click on a link provided by the e-mail to provide their preferences among the available blocks. The priority values are found based on the electronic documents of outpatients in the health care office. So are the original appointments.

To solve the problem expressed by formula (6)–(14) efficiently, the GRA algorithm must be iteratively called because it can only assign one available block to each patient in each iteration, i.e., it can only solve the problem by replacing (13) with (15)

$$\forall i \in \Omega, \sum_{j \in \Pi} x_j(i) \le 1 \tag{15}$$

After each GRA assignment, some available blocks may be still available for those outpatients who have not yet scheduled. We need more GRA processes until no available time slot blocks cover the requests of outpatients. This iteration may affect the global optimization as described by formula (6)–(14), but leads to an efficient solution. That the three optimization goals are synthesized to form one goal is another factor to affect the optimal solution. Note that, the major idea is to transfer some constraints to numbers in order to apply the optimization algorithm. We admit that not all the constraints can be transferred to proper numbers, but there are indeed some constraints that can be processed this way.

V. GROUP ROLE ASSIGNMENT

By E-CARGO [23]–[27], we mean Environments, Classes, Agents, Roles, Groups, and Objects. To deal with the role assignment problems, we emphasize a role set denoted by $\mathcal{A}$ and an agent set denoted by $\mathcal{R}$. Agents in $\mathcal{A}$ are numbered as 0, 1, …, and $m-1$ (m $= |\mathcal{A}|$); and roles in $\mathcal{R}$ are numbered as 0, 1, …, and $n-1$ (n $= |\mathcal{R}|$).

Definition 1: A *role range vector* is a vector of the lower ranges of roles denoted as $L[j] \in \mathcal{N}(0 \le j < n)$.

Definition 2: *A qualification matrix* is defined as $Q : \mathcal{A} \times \mathcal{R} \to [0, 1]$, where, $Q[i, j]$ expresses the qualification value of agent i for role j.

The improvement of the efficiency of the algorithm for the outpatient scheduling problem mainly comes from the formation of the Q matrix.

Definition 3: A *role assignment matrix* is defined as $T : \mathcal{A} \times \mathcal{R} \to \{0, 1\}$. If $T[i, j] = 1$, agent i is assigned to role j and $T[i, j] = 0$ otherwise ($0 \le i < m; 0 \le j < n$). Note that T also expresses a group.

Definition 4: A *group qualification* is defined as the sum of the assigned agents' qualifications, i.e.,$\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} Q[i, j] \times T[i, j]$.

Definition 5: A role r is *workable* if it is assigned enough agents to play it, i.e., $\sum_{i=0}^{m-1} T[i, j] \ge L[j]$ [23]–[27].

Definition 6: A group expressed by T is *workable* if all its roles are workable.

Definition 7: The *group role assignment (GRA) problem*. is to find an assignment matrix T that makes the group qualification the largest, i.e., $\max\{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} Q[ij] \times T[i, j]\}$

$$\text{subject to } \forall j \left(\sum_{i=0}^{m-1} T[i, j] = L[j] \right) (0 \le j < n).$$

VI. FROM THE OUTPATIENT PROBLEM TO THE GROUP ROLE ASSIGNMENT PROBLEM

In terms of GRA, we consider *patients as roles and time slot blocks as agents*. This consideration is explained next.

Definition 8: An *original agent vector* V is an n-vector of the original agent assigned for a role, i.e., $V : \mathcal{R} \to \mathcal{A}$, where $V[j]$ means the agent originally assigned to role j.

We use $F^t(P, i)$ to express the set formed by the elements of vector $P[i]$. If $F^t(P, j) \cap F^t(P, j') \ne \Phi$, we say that roles j and j' are competing on agents (blocks) expressed by $F^t(P, j) \cap F^t(P, j')$. We keep other symbols as described in Sections III and IV.

In solving the outpatient scheduling problem, *the most important step is to build an appropriate Q to reflect the values required in the assignment process*. The vectors W and P and functions g and h can be used to form Q with the following formula ($0 \le i < m, 0 \le j < n$):

$$Q[i, j] = \zeta(P, i, j) \times g(i, j) \times h(i, j) \times \gamma(i) \times \left(\frac{W[j]}{w_{\max}} \right) \tag{16}$$

$$\begin{bmatrix} 0.000 & 0.000 & 0.000 & 0.000 & 0.056 & 0.000 \\ 0.333 & 0.500 & 0.000 & 0.000 & 0.025 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.030 & 0.000 \\ 0.333 & 0.000 & 0.111 & 0.296 & 0.026 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.022 & 0.444 \\ 0.000 & 0.250 & 0.000 & 0.000 & 0.014 & 0.000 \\ 0.167 & 0.000 & 0.056 & 0.148 & 0.011 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.008 & 0.000 \\ 0.167 & 0.000 & 0.056 & 0.148 & 0.011 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.002 & 0.111 \end{bmatrix}$$

Fig. 1. The Q matrix for Table VIII.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 2. The assignment matrix T.

where

$$\zeta(P,i,j) = \begin{cases} 1, & i \in F^t(P,j); \\ 0, & i \notin F^t(P,j); \end{cases} \quad (17)$$

$$\gamma(j) = S'[j]/\max\{S'[0], S'[1], \ldots, S'[n-1]\} \quad (18)$$

Note that, $\zeta(P,i,j)$ tells that time slot blocks (agents) are only qualified for the requested roles (patients), i.e., it is to prevent the situation that an agent (time slot block) is assigned a role (patient) who is not willing to accept; $\gamma(j)$ considers the size of the assigned blocks to patient j and note that we do not use $S'[j]$ directly in order to keep $Q[i,j] \in [0,1]$; $g(i,j)$ in (6) ensures that if two available blocks are patient j's choices, the block with a better preference is assigned, and that if two patients are competing for one time block, the patients' preferences are serialized and $h(i,j)$ in (7) expresses that a time block (agent) has higher qualification if its size fits the time block size better.

From Table VIII, $W = [3,2,3,2,1,3]$, $w_{\max} = 3$, $P = [[1,3,6,8], [1,5], [3,6,8], [3,6,8], [0,1,2,3,4,5,6,7,8,9], [4,9]]$, $S' = [2,3,1,2,1,2]$, $S = [2,4,3,3,3,4,4,4,2,6]$, and $V = [10,11,12,13,14,15]$. For example, $Q[3,2] = 1 \times (1/3) \times (1/3) \times [(3-0)/3] \times (3/3) = 0.111$.

By synthesizing all the above data, we get a qualification matrix Q shown in Fig. 1.

Now, $\mathcal{A}$ is the set of m blocks available at the time of CFC; $\mathcal{R}$ is the set of n patients who have responded to CFC; and Q is an $m \times n$ matrix obtained by formula (16)–(18). Then, the outpatient scheduling problem is, in fact, becoming a GRA problem to find an $m \times n$ assignment matrix T to

$$\max\left\{\sum_{i=0}^{m-1}\sum_{j=0}^{n-1} Q[i,j] \times T[i,j]\right\} \quad (19)$$

subject to

$$\sum_{j=0}^{n-1} T[i,j] \le 1 (0 \le i < m) \quad (20)$$

$$\forall j \left(\sum_{i=0}^{m-1} T[i,j] = L[j]\right) (0 \le j < n) \quad (21)$$

In E-CARGO, that one agent can only play one role in GRA directly follows formula (20). In fact, we can even loosen the restriction in GRA, i.e., remove the requirement of $\forall j, \sum_{i=0}^{m-1} T[i,j] = \mathrm{L}[j] (0 \le j < n)$, because it is acceptable for some bidding outpatients to obtain no block in one round of CFC.

Based on the GRA algorithm, we can obtain matrix T shown as in Fig. 2 for the Q matrix in Fig. 1.

Let $x_j(i)$ in (8) $= T[i,j] (0 \le i < m; 0 \le j < n)$. We obtain Δ_1^* as 26. We can translate the matrix T in Fig. 2 to a list of assignment tuples as the scheduling result of the 1st round CFC (named slots in Table VIII).

Combined with the original agent (time slot block) vector V, the new list of available time slot blocks becomes (shaded blocks Table X). Because all the patients are scheduled, no more iteration of GRA is required. The above list can be used for the 2nd round of CFC. Note that block 10 is formed by combining two original appointments, i.e., (slots 8–9 and slot 10 of Thursday in the third week).

VII. The Algorithm and Complexity

The complexity of the efficient GRA algorithm is polynomial [27]. If we transfer the outpatient scheduling problem into the GRA problem, the outpatient scheduling problem is solved in polynomial time. The following algorithm **OutpatientRescheduling** mainly describes the preprocess, the use of the GRA algorithm, and the post-process.

Note: the following algorithm is described in a Java-like language; "a = b" means to check if a is equivalent to b and "a := b" is to assign the value of b to a.

Input:

Π: A set of outpatients' bidding $\langle \mathrm{x}, \mathrm{y}, \mathrm{Z}, \mathrm{v} \rangle$, where x is the identification of the outpatient; y: the priority value of the outpatient; Z is the list of the bids of time slot blocks of the outpatient, where the position of a block in the list expresses the patient's preference; u is the size of the requested block; and v is the original time slot block of the outpatient.

Ω: A set of the available time slot blocks expressed as $\langle \mathrm{c}', \mathrm{d}', \mathrm{e}', \mathrm{f}' \rangle$, where c$'$ is the starting slot number; d$'$ is size of the block; e$'$ is the day number of the week; and f$'$ is the number of the week.

Output:

χ: A list of tuples $\langle a, b \rangle$ where a means a patient and b means a time slot block.

Ω: a new set of available time slot blocks.

OutpatientRescheduling (Π, Ω, χ)

$\{m_1 := m := |\Omega|;$

$n_1 := n := |\Pi|;$

while ($m_1 > 0$ and $n_1 < n$);

{ **Step 1**:

Transfer Ω into a list $'\Omega$, where $'\Omega[i] \in \Omega$; $(0 \leq i \leq m-1)$;

Step 2:

Transfer Π into 4 lists P, W, C, and V, i.e., $\mathrm{P}[j] := \mathrm{c.x}$, $\mathrm{W}[j] := \mathrm{c.y}$, $\mathrm{C}[j] := \mathrm{c.Z}$, $\mathrm{V}[j] := \mathrm{c.v}(\mathrm{c} \in \Pi, 0 \leq j \leq n-1)$;

Step 3: Note that h, g, Q, ζ, and γ are all $m \times n$ matrices corresponding to formula (6), (7), (16)(18).

$w_{\max:} := \max\{\mathrm{c.y} | (c \in \Pi)\}$;

for $(0 \leq i \leq m-1, 0 \leq j \leq n-1)$ {

if $(i \in C[j])\zeta[i,j] := 1$;

else $\zeta[i,j] := 0$;

if $(S'[j] \leq S[i]) h[i,j] := S'[j]/S[i]$;

else $\mathrm{h}[i,j] := 0$;

$\alpha[j] := C[j].\mathrm{length}$;

$\beta[i,j]$:=index of i in $C[j]$;

$\mathrm{g}[i,j] := (\alpha(C,j)\beta(C,i,j))/\alpha(C,j)$;

$\mathrm{Q}[i,j] := \zeta[i,j] \times \mathrm{h}[i,j] \times \mathrm{g}[i,j] \times \gamma[j] \times \mathrm{W}[j]/\mathrm{w}_{\max}$;

}

Step 4: Note T is an $m \times n$ matrix.

Initialize the assignment matrix T with $\{0\}$;

Call **RatedAssignForOutpatients**(Q, T, m, n);

Step 5: Form the new list of appointments in $\langle \text{patient, time slot block} \rangle$ and adjust the available time slot blocks.

Initialize $\chi[j]$ with $\mathrm{NULL}(0 \leq j \leq n-1)$;

for $(0 \leq i \leq m-1, 0 \leq j \leq n-1)$

if $(\mathrm{T}[i,j] = 1)$ {

$\chi[j] := \langle \mathrm{P}[j], \text{the first P[j].u slots of } '\Omega[i] \rangle$;

if $(\mathrm{P}[j].\mathrm{u} = '\Omega[i].\mathrm{d}')'\Omega[i] := \mathrm{NULL}$;

else { $'\Omega[i].\mathrm{c}' := '\Omega[i].\mathrm{c}' + \mathrm{P}[j].\mathrm{u}$;

$'\Omega[i].\mathrm{d}' := '\Omega[i].\mathrm{d}' - \mathrm{P}[j].\mathrm{u}$; } }

Step 6:

Keep the unscheduled patients the original appointments.

$\Pi := \Phi$;

for $(0 \leq j \leq n-1)$

if $(\chi[j] = \mathrm{NULL})$ {

$\chi[j] := \langle \mathrm{P}[j], \mathrm{V}[j] \rangle$;

$\mathrm{V}[j] := \mathrm{NULL}$;

$\Pi := \Pi \cup \{\mathrm{P}[j]\}$; }

Step 7:

Form the new set of available time slot blocks.

$\Omega := \Phi$;

for $(0 \leq i \leq m-1)$

if $('\Omega[i] \neq \mathrm{NULL})\Omega := \Omega \cap \{'\Omega[i]\}$;

for $(0 \leq j \leq n-1)$

if $(\mathrm{V}[j] \neq \mathrm{NULL})\Omega := \Omega \cap \{\mathrm{V}[j]\}$;

$m_1 := |\Omega|$;

$n_1 := |\Pi|$;

};

if $(n_1 = 0)$ every patient gets an assignment;

else some patients have no assignments;

}//End of **OutpatientRescheduling**

The algorithm **RatedAssignForOutpatients** is described as follows.

Input: Q: *an* $m \times n$ *rated qualification matrix*.

Output: T: *an* $m \times n$ *assignment matrix*.

RatedAssignForOutpatients(Q, T, m, n)

{**Step 1**: $k = max\{m, n\}$;

Transfer the $m \times n$ *matrix* Q to a $k \times k$ matrix M [29];

Step 2: K-M (M); //Call the K-M algorithm;

Step 3: Form the assignment matrix T based on the result of K-M (M);

Step 4: return T;

}

Note that, in the above algorithm, the requirement of $\forall j, \sum_{i=0}^{m-1} T[i,j] = \mathrm{L}[j](0 \leq j < n)$ in the **RatedAssign** algorithm is removed, because it is acceptable for some bidding outpatients to obtain no block in one round of CFC.

Theorem 1: Algorithm **RatedAssignForOutpatients** has the complexity of $O(k^3)$ $(k = max\{m, n\})$.

Proof:

Step 1 has the complexity of $O(k^2)$.

Step 2 **(K-M algorithm)** has $O(k^3)$ [11], [14].

Step 3 has $O(k^2)$.

Step 4 has $O(1)$.

Therefore, the total complexity is $O(k^3)$. ∎

TABLE X
NEW AVAILABLE TIME SLOTS

	M	T	W	T.	F	M	T	W	T.	F	M	T	W	T.	F	M	T	W	T.	F
8:00														10						
9:00	0																			
10:00		2															13			
11:00			3					6												
12:00									8							12				
13:00								7												14
14:00							5		9											
15:00																				
16:00				4																
17:00															11					
18:00	1																			
19:00																				

Theorem 2: Algorithm **OutpatientRescheduling** has the complexity between $O(k^3)$(the best case) and $O(\delta k^3)$ (the worst case), where $k = max\{m, n\}$ and $\delta = min\{m, n\}$.

Proof: Based on Theorem 1, Step 4 has the complexity of $O(k^3)$.

Step 1 has $O(m)$.
Step 2 has $O(n)$.
Step 3 has $O(m \times n)$.
Step 4 has $O(m \times n)$.
Step 5 has $O(n)$.
Step 6 has $O(m + n)$.

Therefore, The total complexity inside the do loop is $O(k^3)$.

As for the *while* loop, the complexity is mainly determined by three factors besides m and n:

1) the original appointments brought in by the bidding outpatients;
2) the conflict bidding for the same blocks;
3) the requested blocks from patients.

The worst situation is that the number of the *do* loops is $\delta = min\{m, n\}$, i.e., all the patients are bidding for the same block that is large enough for all the patients, and each *do* loop satisfies one patient within this block.

The best is within *one* loop:

1) All the patients bid for different blocks and are satisfied in one loop; or
2) All the blocks are assigned to patients in one loop.

In summary, the complexity of algorithm **OutpatientRescheduling** is between $O(k^3)$ and $O(\delta k^3)$. ■

VIII. VERIFICATIONS AND COMPARISONS

To verify the proposed approach, we conducted simulations, performance experimentations, and performance comparisons. The simulations show that the optimality is satisfied. The times used by the proposed algorithm for some random problems are from 0.44 milliseconds (ms) to 2 ms compared with that of ILOG from 280 ms to more than 20 min. The performance experiments show that the time used for large groups ($\mathrm{m} = 80$, $\mathrm{n} = 1000$) is practical, i.e., at most 6.6 s and in average 4.16 s. All the results are included in the supplemental multimedia document.

IX. CONCLUSION AND FUTURE WORK

This paper contributes an efficient approach to outpatient scheduling by a special treatment for collecting patients' choices from available time slots. An exciting future task will be to generalize the proposed approach and to find a way to transform as many constraints as possible into a qualification matrix of GRA. *If all the constraints of a general scheduling problem can be transformed into a qualification matrix of GRA in polynomial time, such a general scheduling problem will be solved within polynomial time*. Such idea may be extended to other scheduling problems [1], [3], [12], [15], [19].

More interest future tasks include: 1) to implement an online service system in real health care environments; 2) to find out an algorithm if functions h and g are correlated; 3) to investigate if the abnormality in Figs. 3 and 4 of the supplement document is a determined phenomenon; 4) to introduce heuristics in solving such assignment problems; and 5) to conduct empirical studies on the proposed approach of CFC.

REFERENCES

[1] B. Alidaee, H. Wang, and F. Landram, "On the flexible demand assignment problems: Case of unmanned aerial vehicles," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 4, pp. 865–868, Oct. 2011.

[2] T. Cayirli and E. Veral, "Outpatient scheduling in health care: A review of literature," *Prod. Oper. Manage.* Jan. 1, 2003. [Online]. Available: http://www.allbusiness.com/health-care/health-care-facilities-clinics/10619562-1.html

[3] C. F. Chu, M. Zhou, H. Chen, and Q. Shen, "A polynomial dynamic programming algorithm for crude oil transportation planning," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 1, pp. 42–55, Jan. 2012.

[4] P. Cramton, Y. Shoham, and R. Steinberg, "Introduction to Combinatorial Auctions," in *Combinatorial Auctions*, P. Cramton, Ed. *et al.* Cambridge, MA: MIT Press, 2006, pp. 1–13.

[5] P. Godin and C. Wang, "Agent-based outpatient scheduling for diagnostic services," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (ICSMC)*, Istanbul, Turkey, 2010, pp. 1851–1856.

[6] S. Gul, B. T. Denton, J. W. Fowler, and T. Huschka, "Bi-criteria scheduling of surgical services for an outpatient procedure center," *Prod. Oper. Manage.*, vol. 20, no. 3, pp. 406–417, May/June 2011.

[7] M. Guo, M. Wagner, and C. West, "Outpatient clinic scheduling—A simulation approach," in *Proc. 36th Winter Simulation Conf.*, Washington, DC, 2004, pp. 1981–1987.

[8] D. Gupta and B. Denton, "Appointment scheduling in health care: Challenges and opportunities," *IIE Trans.*, vol. 40, pp. 800–819, 2008.

[9] IBM, ILOG CPLEX Optimizer. Apr. 2011. [Online]. Available: http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/

[10] G. C. Kaandorp and G. Koole, "Optimal outpatient appointment scheduling," *Health Care Manage. Sci.*, vol. 10, no. 3, pp. 217–229, 2007.

[11] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logist. Quart.*, vol. 2, pp. 83–97, 1955.

[12] H. C. Lau, Z. J. Zhao, S. S. Ge, and T. H. Lee, "Allocating resources in multiagent flowshops with adaptive auctions," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 4, pp. 732–743, Oct. 2011.

[13] N. Liu, S. Ziya, and V. G. Kulkarni, "Dynamic scheduling of outpatient appointments under patient no-shows and cancellations," *Manuf. Service Oper. Manage.*, vol. 12, no. 2, pp. 347–366, Spring, 2010.

[14] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, Mar. 1957.

[15] V. Ng and B. Chan, "Quality assignments for WSDL-based services," in *Proc. Comput. Supported Cooperative Work in Design II*, 2006, vol. 3865, Lecture Notes in Computer Science, pp. 163–173.

[16] J. Patrick and M. L. Puterman, "Improving resource utilization for diagnostic services through flexible inpatient scheduling: A method for improving resource utilization," *J. Oper. Res. Soc.*, vol. 58, pp. 235–245, 2006.

[17] A. Rais and A. Viana, "Operations research in healthcare: A survey," *Int. Trans. Oper. Res.*, vol. 18, pp. 1–31, 2010.

[18] R. L. Rardin, *Optimization in Operations Research*. Upper Saddle River, NJ: Prentice-Hall, 1998.

[19] P. Santibáñez, M. Begen, and D. Atkins, "Surgical block scheduling in a system of hospitals: An application to resource and wait list management in a BC health authority," *Health Care Manage. Sci.*, vol. 10, pp. 269–282, 2007.

[20] I. B. Vermeulen, S. M. Bohte, S. G. Elkhuizen, J. S. Lameris, P. J. M. Bakker, and J. A. La Poutre, "Adaptive resource allocation for efficient patient scheduling," *Artif. Intell. Med.*, vol. 46, no. 1, pp. 67–80, May 2009.

[21] N. Wu and M. C. Zhou, "Schedulability analysis and optimal scheduling of dual-arm cluster tools with residency time constraint and activity time variation," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 1, pp. 203–209, Jan. 2012.

[22] M. B. Wright, "Speeding up the Hungarian algorithm," *Comput. Oper. Res.*, vol. 17, no. 1, pp. 95–96, 1990.

[23] H. Zhu and M. C. Zhou, "Role-based collaboration and its kernel mechanisms," *IEEE Trans. Syst., Man, Cybern.*, vol. 36, no. 4, pt. C, pp. 578–589, Jul. 2006.

[24] H. Zhu and M. C. Zhou, "Role transfer problems and algorithms," *IEEE Trans. Syst., Man, Cybern.*, vol. 36, no. 6, pt. A, pp. 1442–1450, Nov. 2008.

[25] H. Zhu and M. C. Zhou, "M-M role transfer problems and solutions," *IEEE Trans. Syst., Man, Cybern.*, vol. 39, no. 2, pt. A, pp. 448–459, Mar. 2009.

[26] H. Zhu and M. Zhou, "An efficeint solution to the role transfer problem," *IEEE Trans. Syst., Man, Cybern.*, vol. 42, no. 2, pt. A, pp. 491–496, Mar. 2012.

[27] H. Zhu, M. Zhou, and R. Alkins, "Group role assignment via a kuhn-munkres algorithm-based solution," *IEEE Trans. Syst., Man, Cybern.*, vol. 42, no. 3, pt. A, pp. 739–750, 2012.

Haibin Zhu (M'02–SM'04) received the M.S. and Ph.D. degrees in computer science from the National University of Defense Technology, Changsha, China.

He is a Full Professor with the Department of Computer Science and Mathematics, Director and Founder of the Collaborative Systems Laboratory, Nipissing University, Canada. He has published 120+ research papers, four books, and two book chapters. His research interests include collaboration theory, technologies, systems, and applications, human-machine systems, multiagent systems, and distributed intelligent systems.

Dr. Zhu has received many awards. He is serving and served as Co-Chair of the Technical Committee (TC) of Distributed Intelligent Systems of IEEE SMC Society, guest (co-) editor for three special issues of prestigious journals, and organization chairs for many IEEE conferences. He is a founding researcher of Role-Based Collaboration and Adaptive Collaboration. For more information please feel free to browse Dr. Zhu's Website at http://faculty.nipissingu.ca/haibinz/.

Ming Hou (M'05–SM'07) received the Ph.D. degree in human factors engineering from the University of Toronto, Toronto, ON, Canada, in 2002.

He is the Head of the Advanced Interface Group, Defence Research and Development Canada—Toronto, where he is responsible for providing informed decisions to the Canadian Forces on investment in and application of advanced technologies for human-machine interface requirements. His research interests include applied cognition, intelligent adaptive system design, virtual/mixed reality, supervisory control of uninhabited vehicles, and e-learning.

Dr. Hou is a member of the Human Factors and Ergonomics Society and the Association of Computing Machinery. He was the Chair of the Symposium on Human Factors and Ergonomics at the 2009 IEEE Toronto International Conference on Science and Technology for Humanity. He has been the Co-Chair of the International Symposium on Mixed and Virtual Reality since 2004. He is the Canadian National Leader on The Technical Cooperation Program (TTCP) Human Systems Integration Air Technical Panel and the Canadian member of NATO Human Factors Specialist Team under Flight in Non-Segregated Airspace Working Group.

Chun Wang (M'06) received the B.Eng. degree in 1990 from the Huazhong University of Science and Technology, Wuhan, China, and the M.E.Sc. and Ph.D. degrees in computer engineering from the University of Western Ontario, London, ON, Canada, in 2004 and 2008, respectively.

He is currently an Assistant Professor at the Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada. He worked as a Software Engineer and a Project Manager for The China National Petroleum Company from 1990 to 2000. His research focuses on e-Supply Chain, e-Commerce, algorithmic mechanism design, and multiagent systems.

MengChu Zhou (S'88–M'90–SM'93–F'03) received the B.S. degree in electrical engineering from Nanjing University of Science and Technonology, Nanjing, China, in 1983, the M.S. degree in automatic control from the Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, in 1990.

He joined the New Jersey Institute of Technology (NJIT), Newark, NJ, in 1990, and is currently a Professor of Electrical and Computer Engineering and the Director of Discrete-Event Systems Laboratory. He is presently a Professor at Tongji University, Shanghai, China. He has led or participated in projects with total budget over $10 M. He was invited to lecture in many countries and served as a plenary speaker for many international conferences. He has over 460 publications including over 200 journal papers (majority in IEEE TRANSACTIONS. His research interests are in intelligent automation, lifecycle engineering and sustainability evaluation, Petri nets, wireless *ad hoc* and sensor networks, semiconductor manufacturing, Web service, workflow, and energy systems.

Dr. Zhou was the recipient of many prestigious awards. He was recently elevated to Fellow of the American Association for the Advancement of Science (AAAS). He is a founding Editor of the IEEE Press Book Series. He is currently Editor of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, and Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, CYBERNETICS-PART A and the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS. He served as Guest-Editor for many prestigious journals including the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS. He was General Chair for many international conferences. For more information please feel free to browse Dr. Zhou's Website at http://web.njit.edu/~zhou/.