



Centrum voor Wiskunde en Informatica

REPORTRAPPORT

SEN

Software Engineering



Software ENgineering

Improving patient activity schedules by multi-agent Pareto appointment exchanging

I.B. Vermeulen, S.M. Bohte, D.J.A. Somefun,
J.A. La Poutré

REPORT SEN-E0603 APRIL 2006

Centrum voor Wiskunde en Informatica (CWI) is the national research institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organisation for Scientific Research (NWO). CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

Copyright © 2006, Stichting Centrum voor Wiskunde en Informatica
P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-369X

Improving patient activity schedules by multi-agent Pareto appointment exchanging

ABSTRACT

We present a dynamic and distributed approach to the hospital patient scheduling problem: the multi-agent Pareto-improvement appointment exchanging algorithm, MPAEX. It respects the decentralization of scheduling authorities and is capable of continuously adjusting the different patient schedules in response to the dynamic environment. We present models of the hospital patient scheduling problem in terms of the “health care cycle” where a doctor repeatedly orders sets of activities (partial plans) to diagnose and/or treat a patient. We introduce the Theil index to the health care domain to characterize different hospital patient scheduling problems in terms of the degree of relative workload inequality between required resources. In experiments that simulate a broad range of stylized hospital patient scheduling problems, we extensively compare the performance of MPAEX to a set of heuristics. The distributed and dynamic MPAEX has performances almost as good as the best centralized and static scheduling heuristics.

2000 Mathematics Subject Classification: none

1998 ACM Computing Classification System: I.2.11 Distributed Artificial Intelligence

Keywords and Phrases: Multi-agent systems, patient scheduling

Improving Patient Activity Schedules by Multi-agent Pareto Appointment Exchanging

Ivan Vermeulen, Sander Bohte, Koye Somefun, Han La Poutré
CWI, Centre for Mathematics and Computer Science
Amsterdam, The Netherlands
Email: {vermeule, sbohte, koye, hlp}@cwi.nl

Abstract

We present a dynamic and distributed approach to the hospital patient scheduling problem: the multi-agent Pareto-improvement appointment exchanging algorithm, MPAEX. MPAEX is a multi-agent hospital scheduling solution that respects the decentralization of scheduling authorities. MPAEX is capable of continuously adjusting the different patient schedules in response to the dynamic environment. We present models of the hospital patient scheduling problem in terms of the “health care cycle” where a doctor repeatedly orders sets of activities (partial plans) to diagnose and/or treat a patient. We introduce the Theil index to the health care domain to characterize different hospital patient scheduling problems in terms of the degree of relative workload inequality between required resources. In experiments that simulate a broad range of stylized hospital patient scheduling problems, we extensively compare the performance of the MPAEX approach to a set of heuristics. The distributed and dynamic MPAEX has performances almost as good as the best centralized and static scheduling heuristics.

1 Introduction

The first thing that one encounters when seeking medical assistance in a hospital is a schedule: the scheduled medical professionals to consult, time-slots for possible diagnostic or therapeutic machines, and availability of simple resources like examination rooms. Depending on the available capacity, these schedules may be more or less congested. In particular, in countries like The Netherlands and Greece, demand regularly exceeds capacity and substantial waiting lists exist for many medical procedures. Still, even in these situations medical professionals report many schedule inefficiencies. Effective scheduling algorithms should decrease waiting-lists significantly, while increasing hospital efficiency ([1, 2]).

Traditional industrial scheduling techniques are studied in the field of Operations Research (OR). OR techniques are very effective for solving well-defined centralized optimization problems, where the algorithm can determine the optimal schedule for all parties involved. Scheduling solutions based on OR have been implemented in specific targeted health-care problems, like staff planning (for an overview, see [3]). However, OR techniques have so far found little favor

in hospital patient scheduling, in great part due to the inherent distribution of authority in hospitals [4].

In hospital patient scheduling, the scheduling problem is dynamic and always in flux: operations take more or less time than anticipated, crucial staff may not be available, equipment breaks down, and new patients arrive, some with urgent requirements. Not unimportantly, hospitals are also organized around different autonomous departments (wards, ancillary units) each with their own specialty, and each department essentially has authority over their own schedule. Thus, hospital scheduling has strong decentralized characteristics.

The dynamic nature of hospital patient scheduling, together with the decentralization of scheduling authority, suggests that a more suitable approach to hospital patient scheduling may be one that fits the problem domain better: a distributed multi-agent system [5]. Here, we take a first step towards developing a multi-agent hospital agent scheduling solution that respects the current distribution of scheduling authority, and that is capable of continuously adjusting the different schedules in response to the dynamic environment.

The general idea of an agent system [6] is that each party – e.g. doctors, patients, resources – is represented by a software entity – the agent – that acts autonomously on behalf of its owner. Each agent “knows” the preferences and constraints of its owner. The goal of a multi-agent scheduling system is to design the agents and the interaction rules such that together, the agents can arrive at effective schedules.

Related work has studied parts of this multi-agent scheduling problem: Decker & Li [7] consider the problem of resource conflicts in patient scheduling. They design a specific interaction mechanism for resource-representing software agents to prevent such conflicts. The implementation of such a mechanism is shown to achieve substantial productivity gains.

Some distributed constraint solving approaches are also specifically applied to health care scheduling. In [8] resource and patients constraints and preferences (soft constraints) are formally represented and agents use off-the-shelf constraints solvers locally, but a central solver to coordinate constraints is also needed. Additionally, this is a static approach as it only solves an initial scheduling problem and then appointments are assumed to be fixed after that.

Given the distributed and decentralized nature of hospital patient scheduling, the use of a market mechanism for scheduling seems a more natural fit: markets can efficiently distribute scarce resources, they can facilitate dynamic environments, and only price-quotes need to be exchanged between participants, rather than complex constraints and preferences. In [9] a first step is taken in developing a framework for using virtual markets to solve distributed scheduling problems. Different types of auction mechanisms (and additionally bidding strategies [10]) are analyzed. The results show that it is hard to find a general solution. Alternatively in [11] a contracting model for agent-based scheduling is discussed. The search space in this case is very large and a search bias must be set accurately.

A first market-based approach to hospital patient scheduling is taken by Paulussen *et al.* [4]. They introduce software agents that represent the *interests* of the patients, and resources where medical actions take place are represented by resource agents. To distribute the resources amongst these patient agents, Paulussen *et al.* use a market mechanism where patient agents communicate their (private) utility for certain time-slots on a resource via a price mechanism.

Apart from questionable scalability of the system in [4], an additional problem is the use of a utility-function for patient well-being. Clearly, when optimizing a schedule the question is what metric to optimize against in a health-care setting, and patient well-being is an obvious choice [4]. Quantifying *relative* patient well-being however is notoriously hard and any choice will be controversial, with both doctors and patients.

Here, we develop a scheduling method that offers substantial gains without having to consider this difficult issue. We observe that often patients have multiple appointments (e.g. [12]). For instance, a patient may need to get an x-ray and an MRI scan, and then consult with a doctor to discuss the results. We note that the time of the *last* appointment effectively determines the patient's "waiting-time" and a scheduling algorithm can potentially move his/her *other* appointments without any negative effect on the waiting time for the patient.

Guaranteeing "not-worse" for schedule changes means that patients actually have an incentive to cooperate. In any practical implementation of a (re)scheduler, patient cooperation will be essential to make sure that patients are actually willing and able to come to the hospital at the new appointment time.

Being able to guarantee "not-worse" schedule changes thus gives us an opportunity to improve patient waiting times while avoiding the difficult comparison of which patient's well-being benefits how much from an improvement in waiting-time.

In the paper, we present a multi-agent scheduling method that exploits this opportunity by designing agents that exchange appointment times such that no patient is worse off than before, in economics such "nobody-worse" improvement is called a Pareto improvement [13]. Thus, we present a Multi-agent Pareto Appointment EXchanging algorithm: MPAEX.

In the multi-agent system that we develop, patients are assigned an initial schedule for their required treatments. Then, agents acting on behalf of individual patients attempt to exchange the time-slots of the initial appointments with better appointments occupied by other patients. The other patient's agent accepts a proposed exchange of appointments if the resulting schedule is not worse for the patient.

In simulations of (many) hospital patient scheduling problems, we show that when we let patient agents try to improve their patient's schedule, the agents collectively improve the overall patient waiting time.

Based on practical cases, we introduce our hospital patient scheduling model. It represents autonomous departments and resources, as well as individual patients and their activities. We present a semi-dynamic hospital patient scheduling setting to gain fundamental insights and allow comparison with more standard centralized static techniques.

A robust simulation of the distribution of workloads over the various resources in a hospital is a crucial aspect in evaluating scheduling solutions for hospital patient scheduling. For example, the patient scheduling problem may be fundamentally different if either an MRI scan would be a very busy resource, relative to the other resources, or all resources are equally busy. Different hospitals will have different workloads, depending on patient population and the available mix of resources and doctors/staff. To assess accurately how useful different scheduling solutions are, we have to be able to consider a large distribution of different workloads.

As a contribution of this paper, we present a measure for characterizing different workloads: we introduce the Theil index [14] within the hospital scheduling setting. In economics, the Theil index is a common measure of inequality motivated by the notion of entropy. The inequality in a workload can be interpreted as the degree to which bottlenecks are present in the available resources. By conducting experiments for a large range of Theil indices, we obtain a representative sample of the problem space. The characterization of different workloads by their Theil index allows us to compare MPAEX methodically with other (more centralized) scheduling methods.

For these hospital patient scheduling simulations, we find that for initial schedules generated by a pure First-Come-First-Served scheduler, exchanging appointment times results in significant improvements for the final schedule after comprehensive appointment exchanging. For initial schedules with more random initial scheduling (First-Come-Randomly-Served), we obtain even better solutions for the final schedule. This random initial scheduling is introduced to better capture the fact that in current practice, the initial schedule is created in part based on (relatively stochastic) patient and resource availability.

We extensively compare the performance of the decentralized MPAEX approach to a set of centralized heuristics and find that MPAEX performs close to centralized scheduling heuristics. Contrary to these centralized techniques, our approach respects the distributive nature of the scheduling authority, takes patient preferences into account, and is well suited for the dynamic nature of hospital scheduling. Additionally, unlike centralized OR approaches and centralized heuristics, MPAEX is suitable for dynamic environments where patients leave and arrive, and where resources are available or off-line.

Finally the approach explored with MPAEX also provides us with a robust basis: for future research, we want to consider more complex appointment rescheduling environments, the possible inclusion of (artificial) money to enhance exchange possibilities, and more dynamic scheduling settings.

2 Problem

2.1 Hospital Patient Scheduling

Hospitals are increasingly working with databases that record all scheduled activities. The schedules are typically planned within rosters that are predefined by departments or individual doctors, e.g. a doctor sets specific hours for consulting patients, and consultations with patients can only be scheduled in these hours. Authorizations regarding who can access and/or modify the schedules (and rosters) are distributed across the organization depending on local preferences and culture. Most departments are reluctant to allow other departments to make appointments in empty spots in their schedule.

Not surprisingly, many of such electronic scheduling systems are just that: electronic versions of appointment notebooks. Intelligent (re)planning is mostly done by departments, often by hand, and schedule optimization across departments is very hard, as each change needs coordination, usually by phone. Traditional OR techniques are not equipped to deal with coordination between departments with local information, authority and preferences, and fail in such situations. Agent systems are designed for coordination between autonomous

parts with local information.

This current practice of decentralized schedule authorizations in particular leads to complications when a number of appointments across different departments need to be scheduled for a patient. It may very well be that simply switching a scheduled patient in one department to a different time may free up the combination of resources that is needed for another patient. The “switched patient” may not even have a preference for one time or another, but this “free” optimization is currently very hard to achieve. In practice, patients that need resources across departments are not effectively scheduled.

2.2 The Health Care Cycle

To study the hospital patient scheduling problem in detail, we consider the current (typical) health care routine.

Any patient walking into a doctor’s office becomes part of the “health care” cycle: if a real problem is suspected, a number of actions will be scheduled to diagnose the exact nature of the problem and/or a treatment plan is scheduled. Central in the health care cycle is the doctor treating the patient: the doctor first requests a step in diagnostic or treatment options, and upon completion, the results and patient return to the doctor who then determines the next appropriate step. This cycle is also clear in the real medical diagnostic protocols described below; the cycle is illustrated in Figure 1.

Scheduling patient activities is a complex task. A number of diagnostic tests may be needed before a doctor can make a decision about what to do next. Diagnostic tests may require the cooperation of a number of people and resources in the hospital, so appointments have to be scheduled at times when all these resources are available (and the patient of course).

After a consult, a decision is made on what must be done. A plan is made on what activities must take place. We call this a *partial plan*, see Figure 1. A partial plan can consist of a number of different activities, possibly involving different resources and with constraints between them.

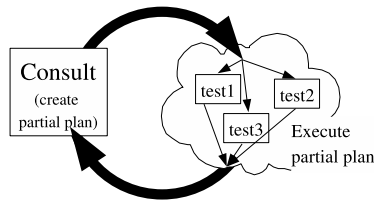


Figure 1: Consult-diagnostic(s)-consult cycle.

2.3 Patient Treatment Plans

The scheduling complexities for typical patient treatment plans that need resources across departments can be illustrated by the workflow in two cases that are typically hard to schedule efficiently. These cases are described in the protocols of the Oncology department of the Academic Medical Centre in Amsterdam.

Diagnosis gallbladder obstruction In Figure 2, the treatment plan for a patient with an obstructed gallbladder pathway is shown [15]. It consists of a

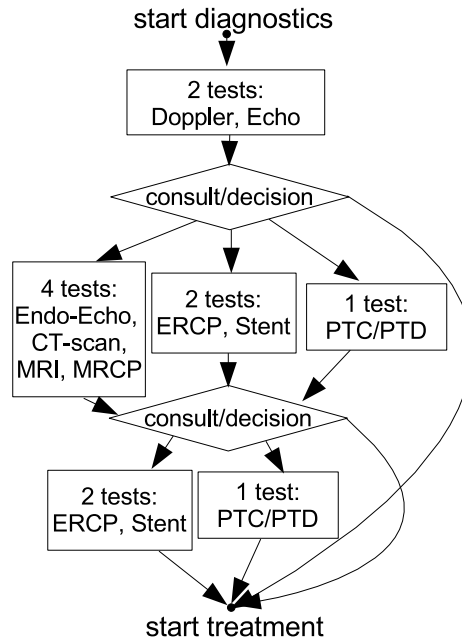


Figure 2: Diagnostical strategy for gallbladder obstruction.

series of successive partial plans where at each point the doctor decides what to do next, based on the diagnostic results. A patient with gallbladder obstruction is first diagnosed with an echo+Doppler for the veins to diagnose for gall stones. If this is the case, treatment can start. Otherwise, to consider the possibility of a tumor, one of three diagnostical plans is selected: either an ERCP+Stent, a PTC/PTD test, or a set of four diagnostics (an endo-echo, a CT-scan, a MRI, and a MRCP) is ordered. From this point again, the patient goes through progressive doctor-diagnosis cycles to determine how to operate.

Diagnosis breast cancer Any persisting breast abnormality requires further diagnostics: the best evaluation of (palpable) mammolesions is through three independent examinations by a surgeon, a radiologist, and a (cyto)pathologist [16]. Each specialist gives an independent assessment that are then equally weighted. If this triple diagnostic suspects a malignant tumor, further (definitive) diagnostics are scheduled, such as a histological or cytological needle biopsy. Upon a definitive positive diagnosis, it is determined how to operate, where further diagnostic tests may be considered depending on the severity of the tumor.

In both cases, the consult-diagnostic(s)-consult cycle is clear; also, in both cases multiple diagnostics are part of a partial plan. Clearly, the workflow cannot progress until all diagnostics are completed.

2.4 Model

We abstract hospitals to a set of resources (including staff), constraints, staff preferences, budget considerations. A hospital is divided into different departments. These departments have a level of autonomy in their actions, and be-

tween them they have common as well as self-interested goals. Departments want to work as efficiently as possible, and they want to take their staff's preferences into account. We model resources on the level of the combined requirements for a single schedule, e.g. an entire ward, or the MRI-schedule including the associated staff etc.

From the medical cases described in the previous subsection, we can distill a number of stylized facts.

Patient Partial Plan With respect to the patient plan, we model a consult as consisting of a limited number of subtasks or activities. These activities can usually be performed in any order, in our model there are no precedence constraints (as in the cases described in Section 2.3). Once all activities have been performed, the responsible doctor determines whether additional activities are necessary. For our approach, it is irrelevant whether activities are new or additional; therefore, we will not make the distinction. Henceforth we will call a number of activities issued at the same time for a single patient by a doctor a partial plan. Based on the result of this partial plan the doctor may issue more activities in the future. We label this new set of activities as a new partial plan. Since most activities within a partial plan are performed on a patient, we assume not more than one activity can be performed at the same time on one patient, hence the different activities in the partial plan have to be scheduled for different times.

Duration of Activities Different activities make use of different resources. Consequently, the standard time, which is reserved for such an activity, may differ from one resource to the other. For example, the time necessary for taking an X-ray or performing an ultrasound examination may differ. We assume that all activities on the same resource take the same amount of time. Between resources, the activity time varies.

Objective The hospital wants to minimize the completion times of patients, given the restriction that within a treatment activities cannot be performed at the same time. To achieve this objective, online decisions about scheduling and especially *rescheduling* are needed that improve the throughput of patients. Completion time of a patient is the time from the creation of the partial plan to completion time of the final activity from this plan.

3 Multi-Agent Pareto Appointment Exchanging

To schedule the patient activities efficiently, we use a distributed approach where software agents interact with each other to exchange appointments where none of two interacting parties is worse off: the agents are *Pareto-improvers*. Worse-off is defined in subjective terms, as each agent acts according to its individual constraints and tries to optimize preferences. Our scheduling method thus amounts to multi-agent Pareto Appointment Exchanging, MPAEX.

In MPAEX, we have two types of agents: resource agents, and patient agents.

Each resource agent represents one resource, and it takes into account constraints like fixed hours, and preferences like preferring not to require over-time. When scheduling an activity to a timeslot on a resource, the resource agent

makes sure no constraints are violated and tries to optimize preferences over resource schedules.

Algorithm 1 Initial timeslot assignment.

For every patient agent:

```
1: while not all activities are scheduled do
2:   select an unscheduled activity from partial plan
3:   ask corresponding resource agent for a timeslot
4:   if the timeslot is not conflicting then
5:     accept timeslot
6:   else
7:     goto 3
```

Algorithm 2 Multi-Agent Pareto Appointment EXchanging (MPAEX)

For every patient agent: (untill no more exchanges can be made)

```
1: select an activity to reschedule
2: ask the corresponding resource agent for a feasible prospective timeslot
3: if there are no alternative prospective timeslots then
4:   goto 1
5: else if the timeslot is empty then
6:   accept and reschedule
7:   goto 1
8: else
9:   get identity of patient agent occupying timeslot
10:  propose the exchange to patient agent occupying timeslot
11:  if the resulting schedule of patient agent occupying timeslot is not worse
    than its current schedule then
12:    both agents reschedule
13:    inform resource agent of exchange
14:    goto 1
15:  else
16:    goto 2
```

Each patient is also represented by a software agent – the patient agent – that has knowledge of the patient’s needs (activities that need to be scheduled, availability) and preferences (when, which doctor, own schedule). We envision that the medical priorities and partial plans in this agent are determined by the consulting doctor, whereas the patient’s preferences, like his/her schedule, are set by the patient.

Patient agents make sure none of their activities overlap and try to get the best possible schedule given the patient’s individual constraints and preferences. To get an initial schedule for their patient, patient agents interact with resource agents to get timeslots for the activities in their partial plan. To improve their schedules, patient agents interact with each other to exchange time-slots. Resource agents do not interact with each other.

In general there are two processes of scheduling: initial timeslot assignment (Algorithm 1), and patient agents improving their schedules by MPAEX (Algorithm 2). In real-life dynamic settings, these processes run together. Patients

arrive one by one and are initially scheduled over the day. Patient agents individually try to improve their patient’s schedule, continuously, or at certain events (such as cancellations, lifted resources constraints, passing deadlines).

Algorithm 1 describes patient agents asking the different resource agents one by one an initial timeslot for the activities of their partial plan. The timeslots cannot be conflicting.

MPAEX implements Algorithm 2: patient agents try to improve their schedule by exchanging appointments for activities. First, the resource agent is contacted to get a feasible prospective timeslot for the selected activity. If this timeslot would improve the schedule but is occupied by another agent, both agents have to agree to exchange the timeslots. No agent will agree to an exchange that will worsen its schedule.

4 Simulating Hospital Patient Scheduling

4.1 Semi-Dynamic Hospital Patient Scheduling

Here, we focus on a semi-dynamic model of hospital patient scheduling. Studying this model first gives us fundamental insights in the performance of our distributed approach relative to alternative approaches. Notably, many state-of-the-art scheduling heuristics are not suitable for fully dynamic and/or distributed scheduling. Furthermore, creating a fully dynamic model requires many ad-hoc decisions, making the models representative only for very specific situations from which it is not straightforward to generalize to other dynamic scheduling instances.

In our simulations, we use “scheduling patients one by one” [17]. This approach formally separates the two processes described in Algorithm 1 and Algorithm 2: we first generate a set of partial plans that have to be scheduled sequentially, one partial plan at a time, without knowledge of what partial plans are to be scheduled next. (This is dynamic, since one patient has to be scheduled before the next one.) Once all partial plans are scheduled, there is a feasible schedule, which is the starting point for MPAEX. Patient agents will then try to improve their schedule (Algorithm 2).

To analyze the different scheduling approaches in a general way, and to compare with centralized and static benchmarks, constraints and preferences in our model are as follows: activities can not overlap on a resource schedule, activities of a partial plan must all be scheduled and can not overlap, patient preferences are such that schedules with earlier completion time (C) are preferred.

We use two fundamental methods for initial timeslot assignment (Algorithm 1): First-Come-First-Served (FCFS), and First-Come-Randomly-Served (FCRS). In FCFS, patients arrive over time and their planned activities are scheduled on the first available time-slot on each required resource. In FCRS, patients arrive over time and their planned activities are scheduled to a random time-slot within a fixed time-window on the required resources¹. Whereas at first sight FCFS seems to relate most closely to current hospital practice, it

¹To determine duration of the time-window, FCRS requires a prediction of the number of activities that can be expected on resources so that after randomly handing out time-slots, the capacity available on the resource will be efficiently used; this data is usually available in hospitals.

ignores the fact that many treatments have limited medical urgency, and the exact date for the appointment is determined both by the patient’s schedule and the first availability of the required resources. To reflect this stochastic element of patient and resource availability, we introduce the FCRS schedule. Note that changes in resource schedules may open up new, better appointment opportunities for patients that fit in their schedule (Algorithm 2), so attempting to reschedule still makes sense. Current practice will fall somewhere between FCFS and FCRS, depending on the actual hospital situation.

Given the patient preferences for the schedule of a partial plan (finish as early as possible) we implement Algorithm 2 in MPAEX as follows: Patient agents always select their last activity for rescheduling. Resources agents will then propose an alternative time-slot, starting from the earliest possible timeslot, and the patient agent will try to exchange his/her time-slot with the patient-agent occupying that time-slot. The deal will be accepted if neither patient is worse off according to their preferences, which here means that completion time will not increase. If not accepted, the patient agent will request another prospective time-slot from the resource agent, and will continue doing so until there are no more prospective time-slots, or a proposed exchange is accepted. The process is repeated for all patient agents iteratively until no exchanges can be made any more.

We found that experiments where an exchange can have more than one activity at a time show very little gain in performance. It is also possible to consider *multilateral* multiple-activities exchanges. The complexity of these types of exchanges quickly becomes intractable. Because of the small complexity of bilateral one-activity appointment exchanges, we only show these results.

4.2 Modeling Hospital Resource Usage

To evaluate the performance of MPAEX fairly, we evaluate hospital patient scheduling performance for a broad distribution of possible hospital characteristics (e.g. [1]).

Two important properties determine the range of possible hospital resource scheduling characteristics: the relative distribution of patients over the resources (how often are patients scheduled for a particular resource workload distribution) and the standardized activity time on the resource, which determines the capacity of the resource. Different combinations can work in favor of one scheduler or the other. Next, we develop a means to methodically vary these parameters such that we can evaluate a representative cross-section of these properties.

Standard Activity Time We model processing time of activities on the same resource to be equal; processing times between resources can vary. Additionally, to capture a wide range of possible activity distributions, we set up four schemes of various standard activity times on the 8 resources. They range from equal activity times on all resources, to large differences: see Table 1.

In the “equal” scheme all activities on all resources take the same time. In the other schemes, activities on some resources take (much) more time than activities on other resources. We experiment with all four different schemes.

Table 1: **Four schemes of standard activity times**

scheme name:	on 8 resources:
equal	1,1,1,1,1,1,1,1
small difference	1,1,2,2,3,3,4,4
reasonable difference	1,2,3,4,5,6,7,8
large difference	1,3,5,7,9,11,13,15

Unequal Workload Distribution In practice not all resources in a hospital are as busy as others, usually there are a limited number of crowded resources. Its number of activities and the standard activity time determine the workload of a resource. By varying the expected number(s) of activities, given a standard activity time scheme, we create instances with varying distributions of workload.

We introduce the use of the Theil index [14] as used in economics, to systematically investigate scheduling performance for various settings of relative resource “busyness”. In our methodology, the Theil index expresses the inequality of the workload distribution. The Theil index is a value between 0 and 1, which is calculated based on the individual workloads of the resources with the formula:

$$T = \frac{1}{m} \sum_j \frac{w_j}{\bar{w}} * \log \frac{w_j}{\bar{w}}, \quad (1)$$

where m is the number of resources, w_j the workload of resource j and \bar{w} the average workload of all resources. This value is a measure of entropy; equal workload corresponds to values near 0, very unequal distributions go towards 1. In Figure 3 we show three distributions of workload of resources (ordered on x-axis) for different Theil indices.

In our experiments in Section 5, instances are created with Theil indices from 0 to 0.6 and ordered in six equal ranges. We will present our results averaged over different runs for each range of Theil indices.

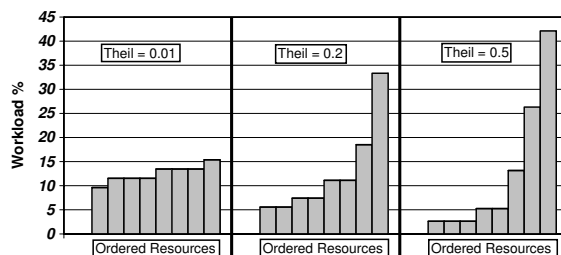


Figure 3: **Examples of relative workload distribution, each bar represents the workload on a resource.**

4.3 Comparison to Centralized Heuristics

We will benchmark the performance of MPAEX relative to centralized and static schedule optimization methods. The semi-dynamical hospital patient scheduling model of Section 4.1 allows us to explicitly compare and contrast with more traditional centralized techniques. These centralized heuristics do not have the

restriction that patients must be scheduled in order of arrival, nor that the final schedule must be a pareto improvement considering all patients.

Scheduling problems can be solved with exact solvers or heuristics [18]. Exact solutions like Branch and Bound (B&B) [19] require exponential time and solving larger practical relevant instances (like 100 patients on 8 resources) optimally is intractable. In practice, we could only run B&B on small problems: i.e., up to 10 patients on 4 resources. Since we are interested in problems of more realistic size, we turn to well established scheduling heuristics to get good performance in reasonable time. From the literature, we use three centralized heuristics: least increment dispatching (**LI**) where priority is based on least increment in the overall objective [20], and two local search algorithms: a hill climber (**HC**) and simulated annealing (**SA**) [21].

All three heuristics (LI, HC, SA) need centralized information. Although that means they are not algorithms applicable in practice, we use them as a comparison for our distributed appointment exchanging approach.

5 Experiments

In this Section, we compare the scheduling performance of our decentralized dynamic MPAEX approach to the centralized static heuristics described in Section 4.3. We use instances of hospital patient schedules as developed in Section 4.1, each instance is characterized by the relative resource workload as discussed in Section 4.2. We generate instances over a broad range of relative resource workload measured by the Theil index.

We took instances with 100 patients having to be scheduled on 8 resources, and where patients have between one and four activities (uniformly distributed) as default. We motivate these numbers from the cases of Section 2.3: patients are usually issued a partial plan with a limited number of activities selected from somewhere between 5 and 15 possible resources (here 8). The number of patients (here 100) is significantly larger than the number of resources ².

In our experiments we study MPAEX with FCFS for initial scheduling: MPAEX(FCFS), as well as with FCFS for initial scheduling: MPAEX(FCFS). We let each scheduling algorithm (FCFS, FCFS, MPAEX(FCFS), MPAEX(FCFS), LI, HC and SA) solve the same generated hospital patient scheduling instance, for a large number of instances.

The overall objective is to minimize the unweighted sum of all individual patients' objectives ($\min \sum_i C_i$). We measure performance for each instance by comparing the schedule result (σ^α) of the different distributed approaches (FCFS, FCFS, MPAEX(FCFS), MPAEX(FCFS)), with the best found schedule (σ^*) from all centralized heuristics (LI, HC, or SA). For most instances, the SA algorithm found the best solution, on a few occasions LI was slightly better. The performance of the distributed approaches relative to the best heuristic solution is computed as σ^*/σ^α per instance and then averaged over 50 instances per Theil index range (for each standard activity time scheme). It means that higher

²The presented results are robust for reasonable variations in the number of patients and resources. Also time complexity of MPAEX scales very well in the number of patients $\leq O(n^2)$, because for each patient agent the first found exchange that is a pareto improvement is accepted.

values represent better performance; the best centralized heuristic performance is set to 1.

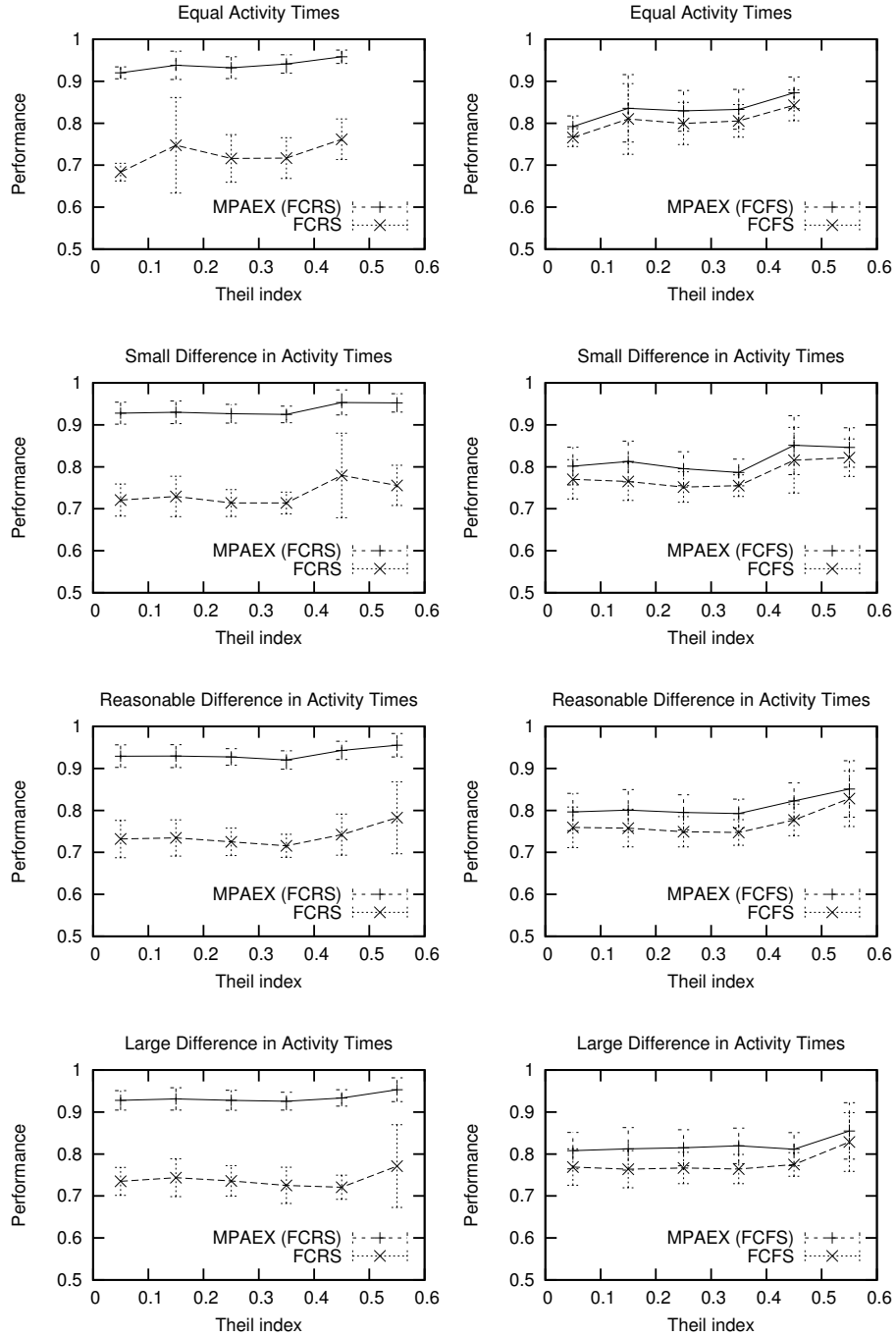


Figure 4: **Relative Performances for different Theil ranges compared to the best centralized static heuristic. Errorbars indicate one standard deviation on either side of the calculated average.**

In Figure 4 we present the average results per Theil range for the four standard activity times schemes of Table 1. We present performance values for the two initial assignment rules (FCFS, FCRS) as well as the improvement after MPAEX (MPAEX(FCFS), MPAEX(FCRS)), relative to centralized scheduling approaches.

The decentralized, dynamic approach of MPAEX performs very close to the best centralized static heuristics. We find that the MPAEX(FCRS) distributed scheduling approach obtains between 92% and 95% of the best performance of the best centralized heuristic (with a typical standard deviation of 2%).

In general, we observe that the results are similar for a wide range of problem instances. For problem instances from a Theil index of 0.4 upwards, the difference between all four approaches and the best heuristic decreases. Only with the scheme of equal activity times and a Theil index between 0 and 0.1 we see a small decrease in performance of the four approaches.

We find that MPAEX(FCRS) outperforms MPAEX(FCFS), although FCFS is the more efficient initial schedule, it also leaves fewer opportunities for MPAEX to improve upon. As the initial FCFS schedule is not that good compared to centralized heuristics, the improve schedule, MPAEX(FCFS) is not that much better. FCRS leaves more improvement opportunities, and the quality of the final schedule MPAEX(FCRS) is (almost) similar to that of the centralized heuristics. In our opinion, this improvement over MPAEX(FCFS) demonstrates the rescheduling quality of the MPAEX approach.

6 Conclusions

We have presented a multi-agent Pareto appointment exchanging algorithm, MPAEX, as a robust, dynamic and distributed solution for patient activity scheduling and rescheduling that actively exploits the Pareto improving scheduling opportunities that are present in hospital patient schedules where patients have to undergo multiple independent activities.

We have presented models of the hospital patient scheduling problem in terms of the “health care cycle” where a doctor repeatedly orders sets of activities (partial plans) to diagnose and/or treat a patient. Additionally, we presented two models for the current practice of initial patient scheduling.

We introduced the Theil index to the hospital patients scheduling domain, to capture the degree of inequality in terms of relative workload between resources that are needed for patient scheduling. In this manner, we demonstrate how a broad range of possible scheduling problems with different relative workloads can be generated.

To compare against existing scheduling solutions, we have presented a stylized semi-dynamical version of the dynamic hospital scheduling problem: first, we schedule patients in order, and then we improve on that schedule with MPAEX.

In experiments over a broad range of such semi-dynamic hospital patient scheduling problems, we show that the MPAEX algorithm arrives at scheduling solutions that are almost as good as centralized solutions.

Unlike centralized heuristics for solving scheduling problems, our multi agent approach can straightforwardly be used in a dynamic environment. Agents can interact concurrently, with local dynamic information (such as: cancellations,

disruptions, expired resource constraints), information does not need to be centrally collected before decisions are made.

In health care preferences on resource utilization (such as: rosters, grouping/alternating similar activities, staff preferences) and of patients (same day appointments, online patient calendars, morning/afternoon preference) are inherently distributed. Multi agent systems can capture such distributed preferences, such that agents act within the rules of scheduling, but try to optimize the schedule for their owner according to the preferences. They will not accept a schedule that is worse than its current one. This is of great value to get collaboration of patients and system acceptance in the hospital.

We have shown results on a semi-dynamic model to provide fundamental insights in the character of hospital patient scheduling and the performance of decentralized dynamic approaches versus traditional centralized static approaches. The setup we developed also provides a proper setting for future approaches and solutions. On the other hand, it is clear that real life problems are fully dynamic. Modeling such fully dynamic scheduling problems involve many more parameters that need to be set and fit to an actual case. In practice, much care has to be taken to obtain generalizable results from such fully dynamic models. In future research we will develop dynamic cases in our studies in actual hospitals.

References

- [1] J. M. Vissers, "Patient flow-based allocation of inpatient resources: A case study," *Europ. J. of Operational Research* 105, 1998.
- [2] C. Marinagi, C. D. Spyropoulos, C. Papatheodorou, and S. Kokkotos, "Continual planning and scheduling for managing patient tests in hospital laboratories," *AI in Medicine* 20(2), 2000.
- [3] C. D. Spyropoulos, "AI planning and scheduling in the medical hospital environment," *AI in Medicine* 20(2), 2000.
- [4] T. O. Paulussen, N. R. Jennings, K. Decker, and A. Heinzl, "Distributed patient scheduling in hospitals," in *Proc. 18th IJCAI*, 2003.
- [5] J. Nealon and A. Moreno, "Agent-based applications in health care," in *Applications of Software Agent Technology in the Health Care Domain*, J. Nealon and A. Moreno, Eds. Birkhueser Verlag, 2003.
- [6] G. Weiss, Ed., *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA: The MIT Press, 1999.
- [7] K. Decker and J. Li, "Coordinating mutually exclusive resources using gpgp," *Proc. 3rd AAMAS*, 2000.
- [8] M. Hannebauer and S. Müller, "Distributed constraint optimization for medical appointment scheduling," in *Proc. 5th Int. Conf. on Autonomous Agents*, 2001.
- [9] M. Wellman, W. Walsh, P. Wurman, and J. MacKie-Mason, "Auction protocols for decentralized scheduling," *Games and Economic Behaviour* 35, 2001.

- [10] D. Reeves, M. Wellman, J. MacKie-Mason, and A. Osepashvili, "Exploring bidding strategies for market-based scheduling," *Decision Support Systems* 39, 2005.
- [11] S. Sen and E. Durfee, "A Contracting Model for Flexible Distributed Scheduling," *Annals of Operations Research*, 1996.
- [12] L. Maruster, T. Weijters, G. de Vries, A. van den Bosch, and W. Daelemans, "Logistic-based patient grouping for multi-disciplinary treatment," *AI in Medicine* 26(1-2), 2002.
- [13] A. Mas-Colell, M. Whinston, and J. R. Green, *Microeconomic Theory*. Oxford University Press, 1995.
- [14] H. Theil, *Economics and Information Theory*. North Holland Publishing Company, 1967.
- [15] "Protocols for gallbladder obstruction," 2005, Academic Medical Centre Amsterdam, Oncology Department, AMC Protocol Document.
- [16] "Protocols for breast cancer," 2005, Academic Medical Centre Amsterdam, Oncology Department, AMC Protocol Document.
- [17] J. Sgall, "On-line scheduling." in *Online Algorithms*, A. Fiat and G. J. Woeginger, Eds. Springer, 1996.
- [18] P. Brucker, *Scheduling Algorithms*. Springer-Verlag, 2001.
- [19] A. H. Land and A. G. Doig, "An automatic method for solving discrete programming problems," *Econometrica*, Vol.28, 1960.
- [20] R. Haupt, "A survey of priority rule-based scheduling," *OR Spektrum*, 11, 1989.
- [21] E. Aarts and J. Lenstra, *Local Search in Combinatorial Optimization*. Princeton University Press, 2003.