# Anatomy of a Scheduling Competition

**Marco Benedetti**
University of Orléans
Orléans, France
*marco.benedetti@univ-orleans.fr*

**Federico Pecora**
Institute for Cognitive Science and Technology
Rome, Italy
*federico.pecora@istc.cnr.it*

**Nicola Policella**
European Space Agency
Darmstadt, Germany
*nicola.policella@esa.int*

**Abstract**

This article analyzes the issues related to designing a regular competitive evaluation for automated scheduling systems. We provide some specific guidelines for designing the competition, and we set out to understand the benefits as well as the drawbacks of its implementation. Both issues are discussed over the backdrop of current relevant competitions in Computer Science.

## Introduction

This article puts forth and attempts to analyze the fundamental issues related to designing a regular competitive evaluation of different approaches to automated scheduling. Scheduling is a multi-faceted discipline, which comprises several distinguishable approaches (an brief overview of which is given in the following section and throughout this article). On account of its manifold nature, the issue of organizing a common forum for comparatively evaluating different approaches must be discussed and collectively agreed upon. Specifically, this article formulates a series of specific questions directed at the scheduling community. Our hope is that collectively answering these questions will lead to a blueprint for a scheduling competition which is well-formed and operative, and which is inclusive with respect to the broad scope of the scheduling community.

The discussion is organized along seven dimensions which we believe are meaningful for designing a competitive evaluation of automated scheduling systems. Throughout the discussion that follows, we summarize how these dimensions are dealt with in current Computer Science (CS) competitions. Over the backdrop of the current competition scenario, we formulate questions regarding the features of the scheduling competition, and provide some possible answers. To conclude, we briefly speculate on the potential benefits and drawbacks of the scheduling competition.

## Different Notions of Scheduling

Broadly speaking, scheduling deals with the allocation of activities (or tasks, jobs) over time. The activities can be modeled as having start times, durations and end times, and can be bound by constraints asserting requirements related to their respective allocation on the time-line. Such constraints can be very different in nature. For instance, they can consist in temporal constraints, asserting anything from simple to generalized precedence relations, i.e., "activity A must finish at least $d$ time units before activity B begins" (minimum time lag), or "activity A must finish no later than $d$ time units after B ends" (maximum time lag). Constraints which affect the allocation of activities in time can also be of a non-temporal nature. For instance, this is the case in project scheduling, where activities need to be sequenced so as to not overload the capacity of one or more resources. A project scheduling problem is identified by means of the resource environment, activity characteristics, and the objective function – scheduling problems are categorized using the three-field notation $\alpha|\beta|\gamma$ (Graham *et al.* 1979).

A somewhat different type of scheduling problem represents several real-word scheduling problems where a solution satisfying all the temporal and resource restrictions may not exist. In particular, problems for which there are typically more requests than can be accommodated with the available resources have been classified as oversubscribed scheduling problems. In this case a solution identifies the best subset of the requests that can be satisfied given the resource and time constraints.

In addition to its theoretical appeal, research in scheduling has also been driven by application. This has brought about the development of numerous automated scheduling systems, some designed to solve a very specific category of problems, others designed to be more general-purpose solving tools. The development of high-performance automated scheduling systems is functional to both application and theoretical research, and different research communities (among which Constraint Programming, Artificial Intelligence, Operations Research and Management Science) have contributed a range of efficient solving techniques as well as their own benchmarks for solver evaluation. Indeed, comparative evaluations of scheduling systems regularly appear in the scheduling literature (e.g., (Beck & Fox 2000; Godard, Laborie, & Nuitjen 2005; Barbulescu *et al.* 2006)), although they are usually conceived as ad-hoc experiments and are limited to the scope of the specific paper.

## Current Competitions and Criteria

In order to establish the issues which need to be taken into account for designing and implementing a scheduling competition, we have analyzed a number of recent competitions in CS. This has lead us to single out the following seven criteria as meaningful aspects underpinning the establishment of a competition. Table 1 shows a comparison along these

dimensions of the competitions we have analyzed.[1]

**Motivation:** the motivation underlying the organization of the competition. On one hand, the existence of a competition may be backed by purely academic motivations, promoting the comparison of specific algorithms, methods or approaches to better understand the theoretical aspects of the computational problem; on the other, the focus may be more on the system as a whole, including aspects such as its fitness for a specific real-world category of problems, its usability, its impact on existing processes, etc., thus indicating a more industry-oriented motivation. The motivational factor is indeed complex and difficult to summarize concisely, and current CS competitions often include other motivations, such as education and dissemination (e.g., RoboCup).

**Participation:** the type and number of participants (data taken from the last edition of the competition). This aspect is important as it indicates whether the current state-of-the-art solving systems are conceived purely for academic evaluation and/or if the technology has industrial potential (respectively, AC and IND in Table 1).

**Benchmarks:** the nature of the benchmark source, namely whether problems are contributed by the participants (CONTRIB), taken from a community-maintained repository (LIBRARY), or disclosed "on-line" during the competition through a competition SERVER.

**Measure:** the evaluation criteria employed to determine system ranking. The overall evaluation criteria is a function $f(m)$, where $m$ is a measure (or combination of measures) belonging to:

- $\sigma$ : the degree to which a system solves the given benchmark(s) (e.g., number of solved problems in the SAT competition).
- $\tau$ : the amount of time taken by the system to complete the benchmark(s) (e.g., the CPU time to find a plan in IPC).
- $\phi$ : a measure related to the quality of solutions found (e.g., the number of satisfied soft constraints in the CSP competition).
- $\omega$ : other measures related to the use of the participating system, such as its ease of use, portability, etc.

Since all competitions we have analyzed impose a form of time-out, solving time also indirectly affects the measure $\sigma$. In order to separate the time-out component from more meaningful temporal measures, we consider time-out as incorporated in the $\sigma$ measure, while $\tau$ refers to more significant measures related to time.

**Disclosure:** the form in which systems participate in the competition. Whether systems need to be completely disclosed in order to participate can be a determining factor in deciding whether or not to submit a system for evaluation, particularly in the case of systems which have strong potential for industrial application. The competitions we have analyzed suggest three "degrees" of disclosure, namely

- SOURCE, where the complete source code is required for submission, and therefore made public;
- BINARY, where source code submission is not required, although binary releases are made public;
- REMOTE, a lesser degree of disclosure where systems are run on participants' computational resources and/or accessed remotely during the competition.

**Knowledge Representation:** the formal representation (or lack thereof) in which competition benchmarks (IN) and competitor results (OUT) are expressed.

**Tracks:** the organization of the competition into tracks. For the purposes of this discussion, we define tracks as competition sub-divisions which are determined by differences in how the problem is approached and/or how the problem is defined.

Clearly, not all of the above criteria are applicable to each competition we have analyzed ("N/A"), nor should this necessarily be the case in the Scheduling Competition.

## Introducing a Scheduling Competition

Why should a scheduling competition be introduced? In our opinion, a more relevant re-statement of this question is

**Q 1** *Should the distributed competition which has been taking place through research papers be given a reference forum?*

In recent years, research in scheduling has made unprecedented advances in the development of techniques that enable better solutions to practical problems. This may lead to think that the scheduling problem is now a solved problem, and that there is no need for scheduling competitions. Unfortunately (and fortunately for the scheduling research community) the previous claim is wrong. We could start by mentioning a consistent number of scheduling problem categories that have been proved to be NP-complete. Much research has gone into approximate algorithms for solving these problems. Yet it is hardly necessary to cite complexity results to substantiate the need for comparing scheduling systems. Even when computational resources are not an issue, there is still the issue of accommodating different classes of constraints and of optimizing under different sets of objective criteria. Moreover, scheduling is rarely a static, well-defined generative task in practice. It is more typically an ongoing, iterative process, situated in a broader planning/problem solving context, and more often than not involving an uncertain executing environment. Each of these additional aspects raises important challenges for scheduling research. Indeed, as asserted in (Smith 2005), the scheduling problem is far from solved.

In this section we further try to motivate the need for a competition in scheduling and we discuss some of the basic underpinnings of such a competition.

### Motivation

In the competitions we have analyzed, it is possible to recognize both commercial and academic motivations.

Competitions such as ICGA, RoboCup and TAC are all backed by important industrial interests. The ICGA accepts participants who have no commercial interest as well as those whose system is or derives from a commercial product. RoboCup is strongly oriented towards showcasing industrial products, and has tapped into the juvenile segment by sponsoring integrated project-oriented education. The trading agent competition (TAC) draws its motivation from e-trading and market analysis, is open to industrial participants, and is overseen by an Industrial Advisory Board.

[1] The number of editions and the number of participants in the last edition of each competition are indicated in parentheses in the first and sixth columns.

| Competition | KR | Bench. | Tracks | Measure | Part. | Disc. | Motivation |
|---|---|---|---|---|---|---|---|
| CASC 1996 (11) | IN, OUT | LIBRARY | 5 divisions, 13 categories | $f(\sigma, \tau, \phi)$ | AC (20) | SOURCE | To stimulate Automated Theorem Proving (ATP) research and system development, and to expose ATP systems within and beyond the ATP community (held in conjunction with CADE). |
| CLIMA 2005 (3) | N/A | N/A | N/A | $f(\sigma, \omega)$ | AC (6) | REMOTE | To stimulate research in the area of multi-agent systems by identifying key problems and collecting suitable benchmarks that can serve as milestones for testing new approaches and techniques from computational logics. |
| CSP 2005 (2) | IN, OUT | LIBRARY, CONTRIB | 5 categories | $f(\sigma, \tau, \phi)$ | AC (21) | BINARY | To improve understanding of the sources of Constraint Satisfaction Problem (CSP) solver efficiency, and the options that should be considered in crafting solvers. |
| GGP 2005 (3) | IN, OUT = {WIN, LOOSE} | SERVER | N/A | $f(\sigma)$ | AC (12) | REMOTE | To assess state-of-the-art in General Game Playing (GGP) systems, i.e., automated systems which can accept a formal description of an arbitrary game and, without further human interaction, can play the game effectively. A $10,000 prize is awarded to the winning team. |
| ICGA 1977 (30) | N/A | N/A | 32 games | $f(\sigma)$ | AC/IND (60) | REMOTE | The International Computer Games Association (ICGA) was founded by computer chess programmers in 1977 to organise championship events for computer programs. The ICGA Tournament aims to facilitate contacts between Computer Science and Commercial Organisations, as well as the International Chess Federation. |
| ICKEPS 2005 (2) | N/A | SERVER | N/A | $f(\sigma, \phi, \omega)$ | AC (7) | REMOTE | To promote the knowledge-based and domain modeling aspects of Planning and Scheduling (P&S), to accelerate knowledge engineering research in AI P&S, to encourage the development and sharing of prototype tools or software platforms that promise more rapid, accessible, and effective ways to construct reliable and efficient P&S systems. |
| IPC 1998 (5) | IN, OUT | LIBRARY | 2 parts, 3 tracks | $f(\sigma, \tau, \phi)$ | AC (12) | BINARY | To provide a forum for empirical comparison of planning systems, to highlight challenges to the community in the form of problems at the edge of current capabilities, to propose new directions for research and to provide a core of common benchmark problems and a representation formalism to aid the comparison and evaluation of planning systems. |
| ITC 2003 (1) | IN, OUT | LIBRARY | N/A | $f(\sigma, \phi)$ | AC (11) | BINARY | The International Timetabling Competition was designed in order to promote research into automated methods for timetabling. It was not designed as a comparison of methods, and discourages drawing strict scientific conclusions from the results. A prize of €300 + free registration to PATAT 2004 was awarded to the winner. |
| PB-Eval 2005 (3) | IN, OUT = {YES, NO, ?} | LIBRARY | N/A | $f(\sigma, \tau, \phi)$ | AC (10) | BINARY | The goal of the Pseudo-Boolean (PB) Evaluation is to assess the state of the art in the field of PB solvers. |
| QBF 2007 (5) | IN, OUT = {YES, NO, ?} | LIBRARY, CONTRIB | 3 tracks | $f(\sigma, \tau)$ | AC (12) | BINARY | Assessing the state of the art in the field of QBF solvers and QBF-based applications. |
| RoboCup 1997 (10) | N/A | N/A | 4 leagues, 7 divisions | $f(\sigma, \omega)$ | AC/IND (440) | BINARY | To foster AI and intelligent robotics research by providing a standard problem where wide range of technologies can be integrated and examined, as well as being used for intergrated project-oriented education. |
| SAT 2002 (5) | IN, OUT = {YES, NO, ?} | LIBRARY, CONTRIB | 3 tracks | $f(\sigma, \tau)$ | AC (47) | SOURCE | To identify new challenging benchmarks and to promote new solvers for the propositional SATisfiability problem (SAT) as well as to compare them with state-of-the-art solvers. |
| SMT-COMP 2005 (3) | IN, OUT = YES, NO, ?} | LIBRARY | 11 divisions | $f(\sigma)$ | AC/IND (12) | BINARY | Push state-of-the-art in Satisfiability Modulo Theories (SMT) for verification applications, in which background theories are used to express verification conditions (e.g., empty theory, real/integer arithmetic, theories of program or hardware structures). |
| TAC 2002 (6) | IN | SERVER | 3 scenarios | $f(\phi)$ | AC/IND (23) | REMOTE | An international forum designed to promote and encourage high quality research into the trading agent problem. |
| TANCS 1999 (2) | N/A | LIBRARY | 4 divisions | $f(\sigma, \tau, \phi)$ | AC (6) | BINARY | To compare the performance of fully automatic, non classical ATP systems (based on tableaux, resolution, rewriting, etc.) in an experimental setting and promote the experimental study on theorem proving and satisfiability testing in non classical logics. |
| Temination 2004 (4) | IN, OUT = {YES, NO, ?} | LIBRARY, CONTRIB | 3 categories | $f(\sigma)$ | AC (6) | BINARY | A competition for termination-proving systems. |

Table 1: A summary along seven dimensions of recent CS competitions.

The large majority of CS competitions also draw their motivation from an academic standpoint. Propositional satisfiability, its quantified generalization, satisfiability modulo theories and constraint satisfaction problems are indeed relevant to some industrial applications, and the competitions organized around these problems (respectively, the SAT, QBF, SMT and CSP competitions shown in Table 1) include limited non-academic participation and/or industrial benchmarks. Nevertheless, the motivation underlying competitive evaluation of automated solvers for these problems stems from the need to push the state of the art in algorithms and/or to promote new challenging benchmarks. In most cases (and in particular for SAT), the competitive context has fostered both theoretical innovation as well as a consistent development effort.

An important feature of scheduling is that its motivations contain a strong component of both factors. On one hand, scheduling techniques have been employed to the advantage of a great deal of real-world problems, e.g., supply-chain management (ILOG April 2007) and production planning (McKay & Wiers 2003). On the other hand, research in scheduling is very much focused on algorithms for solving specific formal problems, such as resource-constrained project scheduling and machine scheduling.

We believe that, in the long run, any successful competition will benefit both commercial and academic interests, regardless of the initial source of motivation. It seems therefore important to assert that industry and academia should both actively participate in the design of the competition, in order to set up the best competition from both perspectives.

**Q 2** *Which are the academic and industrial motivations for organizing a scheduling competition?*

In our opinion, the main motivations of a scheduling com-

petition are the following:

- *Comparing different approaches from different areas*. The scheduling problem has been widely studied by many scientific communities, such as AI, MS, and OR. A scheduling competition is a means to not only evaluate these approaches but also to foster cross-fertilization among these different areas. Ultimately, the goal of the competition should not be limited to proclaiming one or more "winners", rather to achieve a deeper understanding of the scheduling techniques across a wide set of problems.
- *Bridging the gap between scheduling theory and its application in practice*. The development of commercial solvers is often backed by software engineering solutions which facilitate the development of market-grade software products; in contrast, research prototypes are typically aimed at implementing a novel algorithm or approach, and are seldom developed beyond the "demonstrator" level. A competition can contribute to bringing high-quality implementation to research and high-quality research to the commercial realm.
- *Identifying new challenges for scheduling*. New advances from the scheduling research community, and more in general from computer science, will allow to take into account new and more complex scheduling problems.
- *Reducing the fragmentation of research results*. A competitive evaluation can contribute to rationalizing (and avoiding duplication of) research results, leading to a more comprehensive global picture of the field as well as facilitating its assessment.

Overall, a scheduling competition would fill a large void in the current research landscape, namely a rational and ongoing evaluation of state-of-the-art scheduling algorithms with respect to their applicability to different flavors of scheduling. Specifically, the scheduling literature describes scheduling algorithms which are highly tailored to specific types of problems, and the issue of whether these approaches maintain their good qualities in problems that are only slightly different is seldom discussed (Beck, Davenport, & Fox 1997). In this respect, a comparative evaluation on large sets of benchmarks can contribute to presenting a clearer picture of the current state-of-the-art.

## Knowledge Representation

Eight of the competitions we have analyzed provide a completely formalized knowledge representation for the input problem (CASC, CSP, GGP, IPC, ITC, PB-Eval, QBF and SAT). The ability to provide such a clearly defined language for problem specification lies in that fact that the reference problem is described by precise formal attributes, such as variables and constraints in CSP, or CNF formulae in SAT.

Conversely, some competitions have not provided benchmarks expressed in a formal language (TANCS, RoboCup, ICKEPS, ICGA and CLIMA). In all of these cases, the nature of the challenge is such that there is no need to formally describe a benchmark. For instance, in ICGA, the participating systems compete in human-supervised tournaments, in which a programmer is allowed to act as the interface between the game (e.g., the chess board) and the system.

A middle ground among the the competitions we have analyzed is the planning competition (IPC), which represents an interesting anomaly with respect to knowledge representation. A formal specification language for planning domains and problem instances has been established (PDDL (Ghallab *et al.* 1998)) for the purpose of the competition, yet there is no general agreement in the community that the planning problem can be described by this formalism. This has led to successive refinements of the language over the years (Fox & Long 2003; Edelkamp & Hoffmann 2004; Gerevini & Long 2005). Indeed, the planning competition was established through the collaborative deliberations which led to PDDL[2], and has benefited from the relatively narrow forms of planning captured by the initial formulation of the language. Nevertheless, it is also important to mention that PDDL has inevitably biased research in planning, imposing an action-based model as the standard representation for the planning problem.

**Q 3** *Should the scheduling competition provide a single formal language to represent benchmark instances?*

Scheduling poses similar issues in knowledge representation as planning: although there is a general consensus in the community as to the categorization of different forms of scheduling, the nature of these different flavors of scheduling varies quite widely. This is reflected in the presence of a number of different forms of knowledge representation for scheduling problems. One such form is the ProGen representation for Resource-Constrained Project Scheduling problems (Schwindt 1995) (RCPSP), in which the meaningful attributes of a problem are activities, resources, and temporal constraints. The ProGen formalism allows to express (1) the set of activity durations, (2) the temporal constraints which relate their execution with respect to one another, (3) the set of resources and their minimum and maximum available capacity, and (4) the resource requirement of each activity. While it is reasonable to assume that the scheduling competition should compare systems which are tailored to solve these problems, this formalism is not sufficiently expressive for describing other problem categories of interest to the scheduling community.

Other specific representation forms are used to describe similar problems, such as the job-shop problem with due dates and dynamic job arrivals (Demirkol, Mehta, & Uzsoy 1998), or the permutation flow-shop problem (Watson *et al.* 2002). The research carried out on these specific problems has led to the establishment of publicly available benchmarks (Uzsoy 1998; Watson 2002) containing randomly generated problems expressed in a text-based formalism similar to the ProGen format.

The multi-faceted nature of scheduling may suggest that a common representation language for benchmark instances is difficult to achieve. It is true, however, that each of the various forms of scheduling problems that are of interest to the community can be represented by a formal language, and that many of these are sufficiently expressive to represent a number of variants of the scheduling problem. For instance, the ProGen formalism for project scheduling problems can be employed to represent job-shop scheduling problems. Conversely, this formalism does not capture some features of interest to the project scheduling community such as bounded activity durations.

One approach to implementing a knowledge representation infrastructure for the competition is to adopt one or

[2]Mark Boddy, personal communication.

more of these formalisms. This would allow potential participants to benefit from pre-existing, well understood representation which are already supported by many solvers. On the other hand, a limited but more expressive set of especially designed standard specification languages would enable to group systems into clearly defined categories. A versatile formal language for expressing scheduling problems would provide more flexibility in defining tracks, as their definition would not be biased by existing sub-communities, rather on the problems these communities tackle.

In this context, a relevant precedent is given by the IPC. In the case of planning, an entire community chose to describe its problem through a standard language. We believe that the benefits of this approach apply also to scheduling, specifically because the field already has a wealth of widely accepted representation formalisms. In order to avoid the pitfalls of a standard knowledge representation, it is necessary to provide (a) more than one formalism and/or a superset of current formalisms, (b) a formal syntax and semantics, (c) open source parsers and conversion tools, (d) and a regular revision process through which the competition language(s) are updated with respect to new challenges.

## Tracks

Most current CS competitions are in some way subdivided into tracks. The sub-divisions may reflect differences in the benchmarks, as for instance in the SMT-COMP, CSP and TANCS competitions, or in how the benchmark tasks are solved, as for instance in the IPC, where a distinction is also made between optimal and non-optimal solvers. Tracks, in general, are intended to reflect differences in the technology that is showcased through the competition.

**Q 4** *What rationale should be followed to define tracks?*

Where it makes sense to introduce separate tracks depends strongly on the objective of the research community. In the SAT and QBF communities, comparing different approaches is aimed mainly at ascertaining which techniques are more suited for solving a general formulation of the SAT/QBF problem, and differences in the benchmark instances are not considered as meaningful with respect to the products which are showcased. Conversely, the target applications of planning systems are less homogeneous, thus systems compete in tracks in which distinctions are made with respect to both problem and solution characteristics.

A similar line of reasoning can be adopted in the design of a scheduling competition. Current scheduling systems propose solutions for a wide range of problems, some of which differ substantially. Not only are current scheduling systems conceived for different application contexts, but they also present major differences in the type of solution they provide. For instance, RCPSP problems can be solved through a precedence-constraint posting approach (Cesta, Oddi, & Smith 2002), or by means of genetic algorithms (Mendes, Gonçalves, & Resende 2005). The difference between approaches often entails differences in the type of solution which is obtained. In RCPSP, a GA-based approach is likely to obtain solutions faster than a PCP-based method. Conversely, a PCP-based method can carry with it more information, thus what is obtained is not merely an allocation in time of the activities, rather a flexible temporal network which is capable of absorbing (limited) perturbations as a result of polynomial-time constraint propagation. A comparative evaluation of these approaches needs to distinguish the characteristics of the systems with respect to their aim: fast solving vs. information-rich solutions.

**Q 5** *Which tracks should be provided in a scheduling competition?*

Important 'tracks' that clearly emerge in the literature and cannot be ignored are:

- *project scheduling*: problems described by a network of activities, which establishes the temporal relations among the different activities of the problem, and a set of limited capacity resources, which are required in order to execute the different activities.
- *oversubscribed scheduling*: problems consisting of a set of activities which compete for the same set of resources; typically the available resources are not sufficient to satisfy all the activity requests, thus a solution has to sacrifice some activities while preserving schedule quality.
- *on-line scheduling*: problems such as the above where the input instance becomes available over time during solving; solvers thus have to react to new requests (e.g., allocating jobs to machines) with only partial knowledge.
- *schedule execution monitoring*: the problem of maintaining the consistency of a pre-defined schedule during its execution in a real or simulated environment; schedule revision must be quick, and sometimes solution quality must be given secondary priority as the execution of the schedule does not allow for time-intensive computation and/or solution continuity is preferred over drastic on-line changes of the schedule.

The above categories include many different problems. For instance, project scheduling is a generalization of some forms of machine scheduling, where jobs (which are composed of tasks) need to be sequenced so as to not overload the processing capacity of one or more machines.

**Q 6** *Should tracks reflect a more fine-grained distinction among categories, or are the above categories sufficiently representative?*

It may be reasonable to reserve more specific tracks for certain classes of scheduling problems, such as single/multiple machine scheduling, cyclic machine scheduling, job-shops with dynamic job arrivals, flow-shop scheduling, etc. A complementary or additional subdivision can be performed by categorizing problems based on the application they reflect, such as production scheduling, personnel scheduling, university timetabling etc. An application-oriented subdivision contributes to identifying which approaches are more suited for specific applications, an indication which can increase the impact of the scheduling competition.

## Benchmarks

A number of current CS competitions employ randomly-generated benchmarks. Perhaps the most meaningful example is the SAT competition, which stemmed from a scenario similar to that of scheduling today in that comparative SAT solver evaluation was a rather common exercise in the area's literature. In the literature, random 3-SAT benchmarks were perceived to be highly meaningful problem instances for solver comparison. However, while the SAT competition continued the random benchmark tradition, it also added two additional types of benchmarks, namely structured and industrial benchmarks, the difference between the two being

that the former type of problems is obtained as a random problem instance within a well-structured domain, while the latter is a direct SAT formulation of a real-world problem. The inclusion of structured and industrial benchmarks contributed to increasing the understanding of solving approaches by exposing new problem characteristics which are not present in purely random instances.

**Q 7** *Should the competition include randomly-generated benchmarks?*

Indeed, the scheduling community has already equipped itself with both structured and application-derived problems. Specifically, benchmark sets such as the WIOR project scheduling repository (Schwindt 2000) are examples of structured, randomly-generated problem instances (Schwindt 1998), and the previously mentioned permutation flow-shop benchmark set provides both random and structured problem instances.

**Q 8** *Should the competition include benchmarks for dynamic scheduling problems, such as on-line scheduling and schedule execution monitoring?*

In addition to the existence of repositories containing "static" problem instances, the scheduling competition could be designed to take into account the related problem of schedule execution and on-line scheduling. This is the case when activities to be scheduled become known only in proximity of their time of execution, as for example in on-line scheduling (Pruhs, Sgall, & Torng 2004) for multi-user operating systems, web-servers etc. Similar real-time guarantees are required in execution monitoring systems, where an initial schedule (computed off-line) is subject to perturbations (such as delays, resource collapse, etc.) which must be reacted upon while maintaining schedule feasibility. The matter of dynamic execution environments is receiving increasing attention in the literature. Different work has highlighted the issue of taking into account the execution of schedules in unpredictable and uncertain environments (e.g. (Vidal & Ghallab 1996; Hart & Ross 1999; Mc Kay *et al.* 2000; Artigues & Roubellat 2000; Aytug *et al.* 2005; Policella *et al.* 2004; Herroelen & Leus 2004)).

Comparing the performance of such systems can be achieved through an approach to benchmarking similar to some ideas which can be found in the current CS competition landscape. For instance, the TAC competition employs a competition server through which competing systems (travel agents in the TAC Classic game) participate in auctions, or access banking, warehousing and production services (in the TAC Supply Chain Management game). A similar approach is used in the GGP competition, where game rules are transmitted to the players electronically at the beginning of each game (in the GGP game description language) and each participating system must be able to read the rules for each game, receive runtime information from the game manager, and inform the manager of its moves.

Also ICKEPS has employed this mechanism, whereby system evaluation is partially based on problem instances which are accessible through a "simulation server". Although these problem descriptions are given in plain English, access to the specific problem instances occurs through an established communication protocol (over TCP) with the simulation server.

On one hand, the scheduling competition would benefit from the establishment of both a LIBRARY-style set of benchmarks (drawn from the numerous existing benchmark sets) and a SERVER based approach to accommodate the on-line characteristic of some scheduling systems.[3]

On the other hand, this form of evaluation has received less attention in the literature, and is also more challenging to implement than an evaluation of "static" problems. Also, there is more uncertainty as to the number of potential participants proposing such systems, particularly in the case of schedule execution monitoring.

**Q 9** *Are currently available random/structured benchmark libraries sufficient to capture the features of scheduling aplications?*

Indeed, many current applications of scheduling involve very specific real-life problems. These problems include many different types of side-constraints, and are thus seldom instances of well-known problem categories. Because these problems make up a consistent part of applied scheduling research, it is important to include such instances in the competition. If the competition is to drive research in real-life applications of scheduling, a relevant effort should be put into compiling libraries of significant real-world problem instances prior to the competition.

## Evaluation Measures

The issue of how to perform a principled evaluation of systems in current CS competitions has been dealt with in different ways. A first important question that the competition should address is the following:

**Q 10** *Is the scheduling competition aimed at evaluating algorithms, systems, or both?*

As elegantly stated in (Hooker 1994), distinguishing the algorithm from the implementation equates to measuring a phenomenon separately from the apparatus used to investigate it. Indeed, most current CS competitions have, in one way or another, chosen to evaluate both. Evaluation in CASC, CSP, IPC, PB-Eval, SAT, QBF and TANCS depends on the number of solved instances ($\sigma$) as well as CPU time ($\tau$). The strong dependence of algorithm performance on the implementation of the system may indeed compromise the meaningfulness of the evaluation. For instance, system implementation has become a determining factor in recent editions of the SAT competition.

While measuring algorithm performance in terms of number of solved instances and time is not easily measurable in an implementation-independent way, other features of solver performance are less dependent on implementation. The CASC, CSP, ICKEPS, IPC, ITC, PB-Eval, TAC and TANCS competitions all include one or more metrics related to the quality of solutions found, such as the number of violated soft constraints in CSP. In the case of ITC, these measures are strongly related to the real-world nature of the timetabling problem: a solution is evaluated based on how many days a student has only one class, the number of times a student has more than two classes consecutively, or how often a student has class in the last time-slot of the day.

[3] An architecture for benchmarking schedule execution monitoring under controlled uncertainty is described in (Rasconi, Policella, & Cesta 2006).

**Q 11** *Which measures should be adopted in the scheduling competition?*

The scheduling literature points to numerous metrics that can be used to compute the quality of a schedule, some of which are relevant to very specific scheduling problems, others being more generally applicable. Examples relevant for project scheduling include:

- *makespan of a schedule* – the makespan is the latest completion time among all activities in the schedule.
- *schedule fluidity* – the average slack between activities in the schedule, defined as (Cesta, Oddi, & Smith June 1998):

$$fldt = \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{slack(a_i, a_j)}{H \times N \times (n-1)}$$

  where $slack(a_i, a_j)$ is the width of the allowed distance interval between the end time of activity $a_i$ and the start time of activity $a_j$, $H$ is the scheduling horizon, and $N$ is the total number of activities in the schedule.
- *the order strength* – quantifies the effects of the precedence constraints in the schedule (Mastor 1970):

$$OS_P = \frac{|\overline{P}|}{N \times (N-1)/2}$$

  where $N$ is the number of the activities in the schedule, and $\overline{P}$ denotes the set of precedence relations in the transitive closure of the precedence graph.
- *the resource strength* – quantifies the relationship between resource demand and the resource availability in a schedule (Schwindt 1998):

$$RS_k = \frac{C_k^{max} - r_{min}^k}{r_{max}^k - r_{min}^k}$$

  where $r_k$ denotes the $k$-th resource, $r_{min}^k$ is the maximum usage of resource $r_k$ by any activity, and $r_{max}^k$ is the peak demand of resource $r_k$ computed on the earliest start time solution of the infinite capacity version of the problem.

The following are a small sample of metrics which are relevant for on-line scheduling (in addition to makespan and average completion time):

- *flow time metrics* – the flow time of a job $j$ is $F_j = C_j - a_j$, where $C_j$ is the completion time of job $j$ and $a_j$ is the arrival time of the job in an on-line scheduling context. The sum-flow metric is defined as $\sum_j F_j$. The max-flow metric is the maximum value of $F_j$ in the schedule.
- *maximum stretch metric* – the stretch of a job $j$ in a schedule is $strtch_j = \frac{F_j}{p_j}$. The max-stretch metric is defined as the maximum value of $strtch_j$.

In addition, the literature points to metrics aimed at measuring the quality of a schedule with respect to the uncertainty of execution. For instance:

- *disruptibility* – a measure of the stability of a schedule with respect to exogenous events (Policella *et al.* 2004), defined as:

$$dsrp = \frac{1}{N}\sum_{i=1}^{N} \frac{slack_{a_i}}{num_{changes}(a_i, slack_{a_i})}$$

  where $num_{changes}(a_i, slack_{a_i})$ computes the number of activities whose temporal position changes consequently to a delay of size $slack_{a_i}$ imposed on activity $a_i$, and $N$ is the total number of activities in the schedule.

Overall, all the above metrics contribute to defining the $\phi$ dimension in algorithm evaluation. Perhaps more than in other similar competitions, the domain-related nature of the scheduling problem provides a rich set of implementation-independent measures. Interestingly, a rather restricted subset of these measures has been employed in the literature for large-scale comparative evaluations of different approaches. In this respect, a regular competition would enrich the science of scheduling with novel insights and provide a more exhaustive understanding of scheduling approaches.

With respect to the $\sigma$ measure, scheduling presents an interesting difference with some other competitions, in that scheduling systems may have multiple uses. This implies that there are many ways to evaluate the results: we may be interested in the best possible solution given almost unlimited time, or the best solution in a very limited time, or bounds on solution quality. Different techniques apply in these settings, and this can be used to provide criteria for defining tracks which complement or are alternative to the criteria stated previously.

**Q 12** *Should the scheduling competition evaluate system-related features of the participating solvers?*

Few CS competitions have adopted this form of evaluation. CLIMA, ICKEPS and RoboCup are conceived specifically as forums for evaluating complete systems rather than algorithmic approaches, and explicitly take into account other qualities of the participating systems ($\omega$). Nonetheless, comparing systems in addition to algorithms can be an important factor also for the scheduling competition. If the competition aims to bridge the gap between theory and practice in scheduling, then various properties of the system should also be subject to comparison, including usability, impact, etc. It should be said, on the other hand, that any decision in this direction should take care not to intersect with the ICKEPS competition, which focuses on the strongly related problem of knowledge engineering for planning and scheduling.

## Disclosure and Participation

A final remark should be made about the form of participation of competing systems in the scheduling competition.

**Q 13** *To which degree should the scheduling competition require participating systems to be made public (source code, binary or none)?*

The issue here is the following. On one hand, it should be clear which factors contribute to the good performance of an approach. This requires that the algorithmic details are clearly described, but also that they are inspectable, especially given the dependency of algorithm performance on implementation. This would require all participants to submit the source code of their solver. While not even source submission can guarantee scientific validity of the results, guaranteeing the availability of solvers for research purposes after the competition would facilitate cross-fertilization in the community. Indeed, this is the rationale followed in SAT, where all material submitted to the competition is made available to the community.

On the other hand, requiring source code submission may discourage industrial participation. With the exception of CASC and SAT, other CS competitions only require participants to make binaries available. In order to balance the benefits and drawbacks of both approaches, a possible strategy is to allow closed-source submissions to participate hors-concours for the purpose of a more "informal" evaluation, the rationale being that competitive evaluation can only occur if the results benefit the entire scheduling community.

Although open-source distribution may be preferable in some respects, it is important to underscore the added values of disclosing binaries. Namely, a publicly available binary package (a) guarantees that the technology is sufficiently mature to be used by third parties; (b) it enables others to autonomously replicate results and test the scope of applicability of the technology in other domains; (c) it can safeguard against distorted claims, as it implies that anyone contributing a new algorithm must provide a reproducible comparison with relevant solvers; (d) commercially interested parties can evaluate binary prototypes in view of potential further investment in the technology.

## Conclusion

In this article we have put forth some issues related to the establishment of a competitive evaluation of scheduling systems. We have presented a number of questions we feel need to be answered in order to converge towards a scheduling competition. We have presented these arguments against the backdrop of current CS competitions.

There are a number of additional issues we have not elaborated upon which are nevertheless important for establishing the premises of a competition. Among these issues, an important factor is the extent to which the event should push towards competitiveness. The competition should not be overly assertive in "proclaiming winners", rather its outcome should showcase the most successful and novel approaches. The successful implementation of this strategy plays a key role in the success of the competition.

In this paper we have not explicitly focused on potential threats to the scheduling community of a competition. We believe that this issue deserves an in-depth discussion, starting from two key points that have already proved to be meaningful in other CS competitions: first, that a competition which is not sufficiently inclusive can bias research (e.g., the effect of a restricted benchmark specification language); secondly, an excessive focus on incremental details can hinder real progress in the field, a situation which may easily come about if evaluation criteria is not sufficiently implementation-independent.

Although introducing a competition cannot be done without the contribution of the community as a whole, we believe the benefits largely justify the risk of not succeeding. Many fields have benefited from the organization of a competition. The IPC, SAT and QBF competitions, as well as RoboCup in the more distant domain of robotics have fostered measurable advancements in their respective fields. It is likely that a competitive approach to evaluation in a field as fragmented as scheduling could greatly foster cross-fertilization and synergy among researchers with different backgrounds. In addition to the generic benefits a competition can bring to the scientific community, the event can also help to further bridge the gap between theory and practice in scheduling by introducing benchmarks that are grounded in application problems posed by the industry. The feasibility of this applications focus stems from the already strong bias in some areas of scheduling towards industrial problems.

## References


Artigues, C., and Roubellat, F. 2000. A polynomial activity insertion algorithm in a multi-resource schedule with cumulative constraints and multiple modes. *European Journal of Operational Research* 127(2):297–316.

Aytug, H.; Lawley, M. A.; McKay, K. N.; Mohan, S.; and Uzsoy, R. M. 2005. Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal of Operational Research* 165(1):86–110.

Barbulescu, L.; Howe, A. E.; Whitley, L. D.; and Roberts, M. 2006. Understanding Algorithm Performance on an Oversubscribed Scheduling Application. *Journal of Artificial Intelligence Research* 27:577–615.

Beck, J., and Fox, M. 2000. Constraint-directed techniques for scheduling alternative activities. *Artificial Intelligence* 121:211–250.

Beck, J.; Davenport, A.; and Fox, M. 1997. Five Pitfalls of Empirical Scheduling Research. In *Proceedings of 3rd International Conference on Principles and Practice of Constraint Programming (CP'97)*, number 1330 in Lecture Notes in Computer Science, 390–404. Springer.

Cesta, A.; Oddi, A.; and Smith, S. 2002. A Constrained-Based Method for Project Scheduling with Time Windows. *Journal of Heuristics* 8(1):109–135.

Cesta, A.; Oddi, A.; and Smith, S. June, 1998. Profile-Based Algorithms to Solve Multi-Capacitated Metric Scheduling Problems. In *Proceedings of the 5th International Conference on Artificial Intelligence Planning Systems*.

Demirkol, E.; Mehta, S.; and Uzsoy, R. 1998. Benchmarks for Shop Scheduling Problems. *European Journal of Operational Research* 109(1):137–141.

Edelkamp, S., and Hoffmann, J. 2004. PDDL2.2: The Language for the Classical Part of the 4th International Planning Competition. Technical report, Technical Report 195 Computer Science Department, University of Freiburg.

Fox, M., and Long, D. 2003. PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. *Journal of Artificial Intelligence Research* 20:61–124.

Gerevini, A., and Long, D. 2005. Plan Constraints and Preferences in PDDL3. Technical report, Dept. of Electronics for Automation, University of Brescia, Italy.

Ghallab, M.; Howe, A.; Knoblock, C.; McDermott, D.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. PDDL — The Planning Domain Definition Language, AIPS 98 Planning Competition Committee.

Godard, D.; Laborie, P.; and Nuitjen, W. 2005. Randomized Large Neighborhood Search for Cumulative Scheduling. In *Proceedings of the International Conference on Automated Planning & Scheduling (ICAPS 2005)*, 81–89.

Graham, R.; Lawler, E.; Lenstra, J.; and Rinnooy Kan, A. 1979. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Ann. Discrete Math.* (4):287–326.

Hart, E., and Ross, P. 1999. An immune system approach to scheduling in changing environments. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-99*, 1559–1565. Morgan Kaufman.

Herroelen, W., and Leus, R. 2004. The construction of stable project baseline schedules. *European Journal of Operational Research* 156:550–565.

Hooker, J. 1994. Needed: An empirical science of algorithms. *Operations Research* 42(2):201–212.

ILOG. April 2007. White Paper: Improving Plant Performance and Flexibility in Process Manufacturing.

Mastor, A. 1970. An Experimental and Comparative Evaluation of Production Line Balancing Tehniques. *Management Science* 16:728–746.

Mc Kay, K. N.; Morton, T. E.; Ramnath, P.; and Wang, J. 2000. Aversion dynamics scheduling when the system changes. *Journal of Scheduling* 3(2):71–88.

McKay, K., and Wiers, V. 2003. Integrated Decision Support for Planning, Scheduling, and Dispatching Tasks in a Focused Factory. *Computers In Industry* 1(50):5–14.

Mendes, J.; Gonçalves, J.; and Resende, M. 2005. A Random Key Based Genetic Algorithm for the Resource Constrained Project Scheduling Problem. Technical report, AT&T Labs. AT&T Labs Research Technical Report TD-6DUK2C.

Policella, N.; Smith, S. F.; Cesta, A.; and Oddi, A. 2004. Generating Robust Schedules through Temporal Flexibility. In *Proceedings of the 14th International Conference on Automated Planning & Scheduling, ICAPS'04*, 209–218. AAAI.

Pruhs, K.; Sgall, J.; and Torng, E. 2004. Online scheduling. In *Handbook on Scheduling*. CRC Press.

Rasconi, R.; Policella, N.; and Cesta, A. 2006. Fix the Schedule or Solve Again? Comparing Constraint-Based Approaches to Schedule Execution. In *Proceedings of the ICAPS Workshop on Constraint Satisfaction Techniques for Planning and Scheduling Problems*.

Schwindt, C. 1995. Project Generator ProGen/max and PSP/max-library, University Karlsruhe, Institute for Economic Theory and Operations Research.

Schwindt, C. 1998. Generation of Resource-Constrained Project Scheduling Problems Subject to Temporal Constraints. Report WIOR-543, Universität Karlsruhe.

Schwindt, C. 2000. WIOR Project Scheduling benchmark website: `http://www.wior.uni-karlsruhe.de/LS_Neumann/Forschung/ProGenMax/index_html`. Accessed April, 2007.

Smith, S. F. 2005. Is Scheduling a Solved Problem? In G. Kendall, A. Burke, S. Petrovic, and M. Gendreau., ed., *Multidisciplinary Scheduling: Theory and Application*. Springer. 3–18.

Uzsoy, R. 1998. Purdue Electronics Manufacturing Research Group repository: `http://cobweb.ecn.purdue.edu/~uzsoy2/Problems/main.html`. Accessed April, 2007.

Vidal, T., and Ghallab, M. 1996. Dealing with uncertain durations in temporal constraint networks dedicated to planning. In *Proceedings of the 12th European Conference on Artificial Intelligence*, 48–52.

Watson, J.; Barbulescu, L.; Whitley, L.; and Howe, A. 2002. Contrasting Structured and Random Permutation Flow-Shop Scheduling Problems: Search-Space Topology and Algorithm Performance. *Informs Journal on Computing* 14(2):98–123.

Watson, J.-P. 2002. Structured versus Random Benchmarks for the Permutation Flow-Shop Scheduling Problem: `http://www.cs.colostate.edu/sched/generatorNew/index.html`. Accessed April, 2007.