J Sched (2016) 19:271–283
DOI 10.1007/s10951-014-0402-0

The Multi-Mode Resource-Constrained Multi-Project Scheduling Problem

The MISTA 2013 challenge

Tony Wauters · Joris Kinable · Pieter Smet · Wim Vancroonenburg · Greet Vanden Berghe · Jannes Verstichel

Received: 30 November 2013 / Accepted: 28 October 2014 / Published online: 22 November 2014

Abstract Scheduling projects is a difficult and time consuming process, and has far-reaching implications for any organization's operations. By generalizing various aspects of project scheduling, decision makers are enabled to capture reality and act accordingly. In the context of the MISTA 2013 conference, the first MISTA challenge, organized by the authors, introduced such a general problem model: the Multi-Mode Resource-Constrained Multi-Project Scheduling Problem (MRCMPSP). The present paper reports on the competition and provides a discussion on its results. Furthermore, it provides an analysis of the submitted algorithms, and a study of their common elements. By making all benchmark datasets and results publicly available, further research on the MRCMPSP is stimulated.

Keywords Project scheduling · Multi-mode · Multi-project · Heuristics · Competition

1 Introduction

Project scheduling is well known in operations research and management, and has been the subject of intensive research since the late fifties. It has seen many practical applications, ranging from high-level scheduling of software projects and construction projects to fine-grained scheduling of machine operations on a production floor [e.g. scheduling orders in a food production facility (Wauters et al. 2012)]. However, the classical Resource-Constrained Project Scheduling Problem (RCPSP) is not general enough to model all aspects of a real-world problem. Therefore, many extensions to the RCPSP have been presented in the literature, such as min/max time lags, multi-mode, multi-skill, etc.

The MISTA 2013 challenge, organized along with the MISTA 2013 conference, tries to encourage research on a new general project scheduling problem: the Multi-Mode Resource-Constrained Multi-Project Scheduling Problem (MRCMPSP). Multiple projects have to be scheduled simultaneously, while taking into account the availability of local and global resources. The MRCMPSP has a considerable practical relevance, especially in the construction and production sectors. The challenge results in a new benchmark for the MRCMPSP by releasing public instances, a solution validator, an instance generator and state-of-the-art results. The MRCMPSP is attractive for its simple problem description and the lack of complicated constraints to express and evaluate. It therefore requires relatively limited modelling effort, which enables competitors and future researchers to focus on developing powerful algorithms and on the intrinsic scheduling complexity. The MISTA 2013 challenge is, as far as the authors know, the first academic project scheduling competition. A large international audience has been reached (Fig. 1): 21 teams from 14 countries and 3 continents submitted algorithms. Teams were compared on a benchmark PC with a fixed time limit.

Challenges/competitions are important for stimulating research on particular problems or problem domains. Examples of such challenges are the international timetabling competitions (ITC 2002, ITC 2007, ITC 2011) (McCollum et al. 2007; Post et al. 2013), the nurse rostering competition (Haspeslagh et al. 2012), the ROADEF challenge series (Solnon et al. 2008) and the cross-domain heuristic search challenge (CHeSC 2011) (Burke et al. 2011).

T. Wauters (✉) · J. Kinable · P. Smet · W. Vancroonenburg · G. Vanden Berghe · J. Verstichel
Department of Computer Science, CODeS-iMinds-ITEC, KU Leuven, Gebroeders De Smetstraat 1, 9000 Gent, Belgium
e-mail: tony.wauters@cs.kuleuven.be

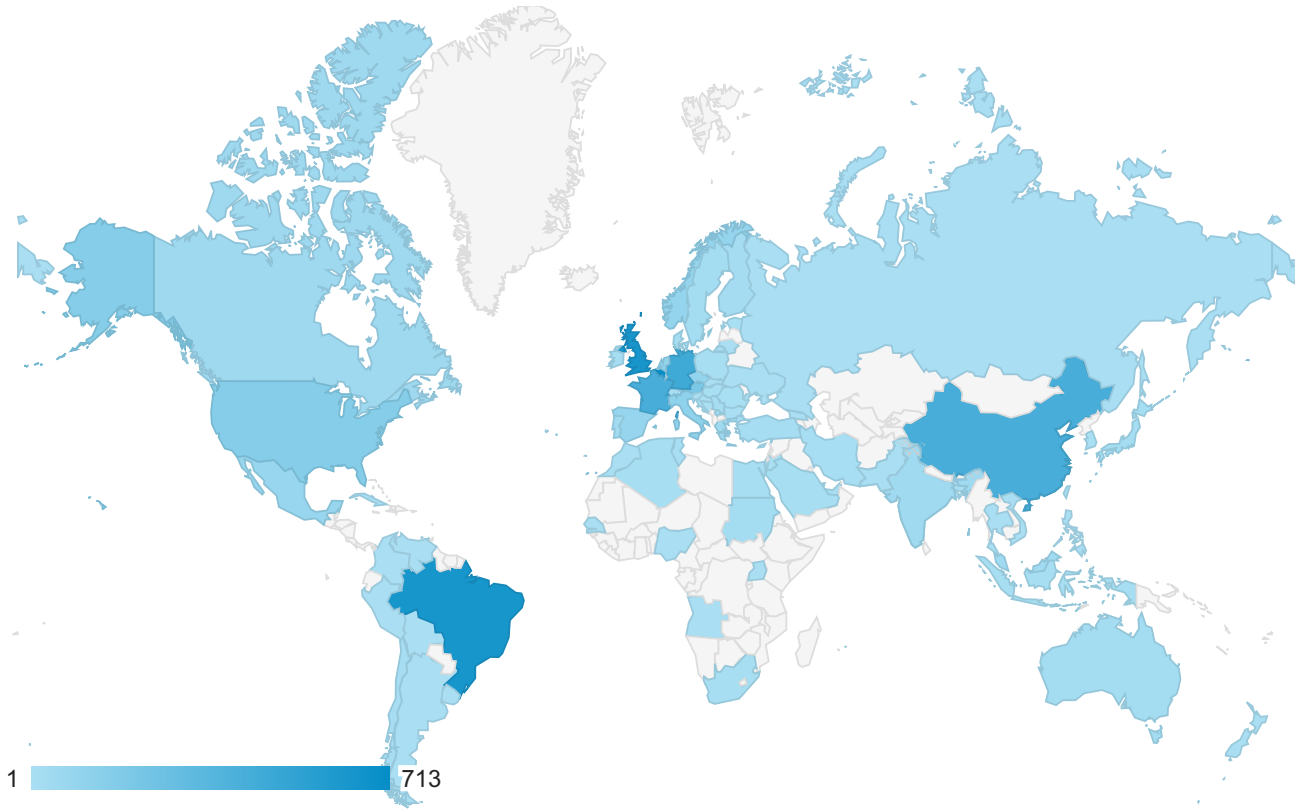

Fig. 1 World map of MISTA challenge website visits by country (image courtesy of Google Analytics)

The present paper is structured as follows: Section 2 introduces the MRCPSP and gives a formal definition of the problem, its constraints and objectives. Section 3 describes the competition format, the different phases, test setup and final ranking. Section 4 provides a detailed analysis and discussion of the competition results and submitted algorithms. A conclusion and future perspectives are given in Sect. 5.

2 Problem description

The MRCMPSP is a generalization of the (single-mode) RCPSP in two ways (Fig. 2). First, jobs can be executed in multiple modes (MRCPSP), and second, multiple projects have to be scheduled simultaneously while sharing resources (RCMPSP). These modes allow including time/cost, time/resource and resource/resource trade-offs. The RCPSP was proven to be NP-hard (Blazewicz et al. 1983), consequently the MRCMPSP is also NP-hard. Moreover, the problem of finding a feasible mode assignment subject to more than one non-renewable resource constraint is NP-complete (Kolisch and Drexl 1997).

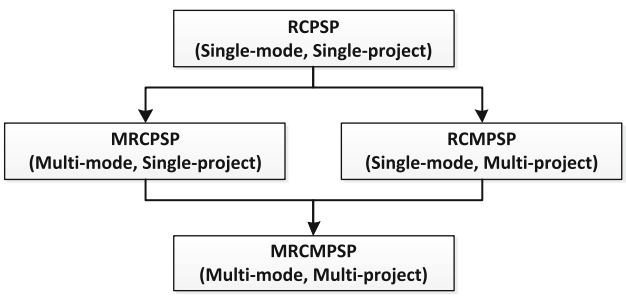

Fig. 2 RCPSP generalization hierarchy

2.1 Projects and activities

A set $\mathscr{P}$ of n projects has to be scheduled, under the restriction of several time and resource constraints. The projects are identified by their index $i \in \mathscr{P}$, with $\mathscr{P} = \{0, \ldots, n-1\}$. Each project $i \in \mathscr{P}$ consists of a set of non-preemptive jobs or activities J_i. For each activity $j \in \{1, \ldots, |J_i|\}$ of project i, a start time $s_{ij} \geq 0$ has to be determined. The first and last activities of the projects ($j = 0$, resp. $j = |J_i| + 1$) are dummy activities, which have a zero duration and no resource requirements. Each project $i \in \mathscr{P}$ has a release date r_i, i.e. the earliest time at which the activities of project i can start ($s_{i0} \geq r_i$).

2.2 Local and global resources

A set $L_i = L_i^\rho \cup L_i^\nu$ of resources is associated with each project $i \in \mathscr{P}$, where the subset $L_i^\rho = \{1, \ldots, |L_i^\rho|\}$ denotes the local renewable resources and $L_i^\nu = \{|L_i^\rho| + 1, \ldots, |L_i^\rho| + |L_i^\nu|\}$ the non-renewable resources. Renewable resources have a fixed capacity per time unit, while the non-renewable resources have a fixed capacity over the entire project duration. Each resource $l \in L_i$ has a fixed capacity of c_{il}.

Finally, a set G^ρ of global renewable resources is shared among the projects. Similar to the local resources, the availability of the global resources is limited by c_g^ρ, $g \in G^\rho$. There are no global non-renewable resources.

2.3 Execution modes

For each activity $j \in J_i$, $i \in \mathscr{P}$, one or more execution modes are available. The execution mode in which an activity is performed determines both the time required to complete the activity, as well as the specific resource requirements. Let $M_{ij} = \{1, \ldots, |M_{ij}|\}$ be the set of possible modes in which activity $j \in J_i$, $i \in \mathscr{P}$, can be performed, and d_{ijm} the processing time of activity $j \in J_i$ in mode $m \in M_{ij}$. The constants r_{ijml}^ρ, r_{ijml}^ν and r_{ijmg}^ρ, respectively, define the consumption of local renewable, non-renewable and global renewable resources when activity $j \in J_i$ is processed in mode $m \in M_{ij}$.

Note that a feasible schedule must always adhere to the following resource constraints:

$$\sum_{j \in J_i} \sum_{m \in M_{ij}} x_{ijt}\, y_{ijm}\, r_{ijml}^\rho \le c_{il} \qquad \forall i \in \mathscr{P}, l \in L_i^\rho, t \in [0, T] \tag{1}$$

$$\sum_{j \in J_i} \sum_{m \in M_{ij}} y_{ijm}\, r_{ijml}^\nu \le c_{il} \qquad \forall i \in \mathscr{P}, l \in L_i^\nu \tag{2}$$

$$\sum_{i \in \mathscr{P}} \sum_{j \in J_i} \sum_{m \in M_{ij}} x_{ijt} y_{ijm}\, r_{ijmg}^\rho \le c_g \quad \forall g \in G^\rho, t \in [0, T] \tag{3}$$

$$\sum_{m \in M_{ij}} y_{ijm} = 1 \qquad \forall i \in \mathscr{P}, j \in J_i \tag{4}$$

$$x_{ijt} \in \{0, 1\} \qquad \forall i \in \mathscr{P}, j \in J_i, m \in M_{ij}, t \in [0, T] \tag{5}$$

$$y_{ijm} \in \{0, 1\} \qquad \forall i \in \mathscr{P}, j \in J_i, m \in M_{ij} \tag{6}$$

where binary variables x_{ijt} indicate whether a particular activity $j \in J_i$ is performed at time $t \in [0, T]$, while binary variables y_{ijm} denote whether activity $j \in J_i$ runs in mode $m \in M_{ij}$. T is an upper bound on the time horizon of the scheduling problem.

2.4 Precedence constraints

Certain activities may require the completion of other tasks prior to their own start. In such a case, if activity $j \in J_i$ must precede activity $j' \in J_i$, a precedence relation $j \prec j'$ is defined. Feasible schedules must obey all precedence relations:

$$s_{ij} + \sum_{m \in M_{ij}} y_{ijm} d_{ijm} \le s_{ij'} \qquad if\ j \prec j' \tag{7}$$

$pred(j)$ denotes the predecessor set of activity j, i.e. $pred(j) = \{j' | j' \prec j\}$.

2.5 Objectives

The objective is to find a feasible schedule while minimizing the total project delay (TPD) (Sect. 2.5.1) and the total makespan (TMS) (Sect. 2.5.2). The two objectives are considered in lexicographical order, where TPD is the primary objective, while TMS is used as a tie breaker.

2.5.1 Total project delay (TPD)

The project delay is defined as the difference between the critical path duration (CPD), a theoretical lower bound on the earliest finish time of the project, and the actual project duration (makespan). More formally, the project delay ($\mathrm{PD_i}$) for a project $i \in \mathscr{P}$ is defined as:

$$\mathrm{PD}_i = \mathrm{MS}_i - \mathrm{CPD}_i$$

where MS_i, the makespan of project i, is:

$$\mathrm{MS}_i = f_i - r_i \quad \text{with } f_i = s_{i(|J_i|+1)}.$$

and CPD_i, the critical path duration[1] of project i, is:

$$\mathrm{CPD}_i = \mathrm{EF}_{i(|J_i|+1)},$$

where EF_{ij} is the earliest finish time of activity $j \in J_i$, which is recursively defined as:

$$\mathrm{EF}_{i0} = 0 \qquad \forall i \in \mathscr{P}$$
$$\mathrm{EF}_{ij} = \max_{j' \in pred(j)} (EF_{ij'}) + \min_m (d_{ijm}) \qquad \forall i \in \mathscr{P}, j > 0$$

[1] Values for the critical path duration of the projects are provided in the input file.

Finally, the TPD is expressed as:

$$\text{TPD} = \sum_{i=1}^{n} \text{PD}_i$$

2.5.2 Total makespan (TMS)

The TMS is the duration of the whole multi-project schedule.

$$TMS = \max_{i \in \mathscr{P}}(f_i) - \min_{i \in \mathscr{P}}(r_i)$$

3 Competition

The initial announcement of the MISTA 2013 challenge attracted 21 teams to register for the competition. The majority of participants were affiliated with academic institutions, whereas only three teams were affiliated with industry. The competition's rules did not stipulate any restrictions on how an algorithm could obtain a solution. The only requirements for the approach were that the code was executable by the organizers, and that it was either completely free or free under academic licencing.

The competition consisted of three phases. The teams were provided with a first dataset to guide the initial development of their algorithms (A instances). Problem size in the first dataset ranged from 2 projects and 10 activities per project to 10 projects and 30 activities per project. Three months after the competition's announcement, the teams were asked to submit their approach in order to determine the qualifying teams for the next phase. The qualification criteria consisted of executable software that was able to produce feasible solutions for the initial dataset within a time limit of 5 min. 16 out of the 21 registered teams qualified for the second phase.

After the announcement of the qualification results, a second set, containing larger instances, was published (B instances). The problem size varied between instances with 10 projects and 10 activities per project and instances with 20 projects and 30 activities per project. Again, the participants had 3 months to fine-tune their approaches, given the new dataset and the best obtained solutions from the qualifications phase. At the end of this second phase, 9 out of 16 teams submitted an algorithm for the final phase.

In the last phase, the performance of the final submissions was evaluated. The final ranking was determined by running each participant's algorithm ten times on all instances from the second dataset and an additional hidden dataset (X instances), with instances of similar size as the B instances. These experiments were executed by the organizers on a recent desktop pc[2], with a time limit[3] of 5 min per instance. For each instance, the algorithms were ranked based on the average of all runs. Finally, the ranks were averaged over all instances per team. Table 1 shows the final results.

Table 1 Final ranking

Rank	Team ID	Members	Mean rank
1	11	Asta et al. (2013)	1.10
2	8	Geiger (2013)	2.55
3	1	Toffolo et al. (2013)	3.05
4	20	Artigues and Hebrard (2013)	3.60
5	15	Alonso-Pecina et al. (2013)	6.75
5	13	Borba et al. (2013)	6.75
5	17	López-Ibáñez et al. (2013)	6.75
8	14	Bouly et al. (2013)	6.85
9	21	Schnell (2013)	7.60

Additional details on the competition rules, the datasets and general information regarding the competition can be found at the competition web page (Wauters et al. 2013).

3.1 Instance generation

Instances of this new MRCMPSP have been generated by combining multiple MRCPSP instances from PSPLIB. Release dates and global resources had to be added. The release dates are generated using a poisson process, i.e. the project inter arrival times are exponentially distributed with $\lambda = 0.2$. The release date of the first project is always $r_0 = 0$. One of the two renewable resources is a global resource with a 2/3 probability. The capacity of the global renewable resources has been set to a value between 50 and 100% of the local renewable resource capacity. To guarantee feasibility the capacity should be at least 10, i.e. in the PSPLib files, there are jobs with a resource consumption of 10. The following formula is applied:

$$c_g^{\rho} = \max\left(CAP_{\max}\frac{5 + rand(6)}{10}, 10\right),$$

where $CAP_{\max}$ is the maximum capacity of the corresponding local renewable resources and $rand(6)$ is a uniform random number between 0 (inclusive) and 6 (exclusive). The instance generator is available for download on the competition web page.

[2] Intel Core i7-2600 at 3.4 Ghz and 8 GB RAM with hyper-threading, turbo boost and energy saver disabled. ITC2011 benchmark tool score: 619 s.

[3] The time limit was checked by measuring the wall clock time of the algorithms. The algorithm should stop autonomously within the time limit. Small deviations of less than 0.1 s were allowed.

4 Discussion

4.1 Detailed results

Table 2 gives an overview of the detailed results per team and per instance, averaged over 10 runs. This produced the final ranking (by mean rank over all instances) shown in Table 3. It is clear that the competition's final winner, team 11, produced the best average results over all instances, except for instances B-2 and X-1. The second and third ranking positions were more contested, with team 1, team 8 and team 20 alternatingly producing second and third best results (Table 4).

Figure 3 shows box plots of each team's results per instance, giving an indication of the spread of the results. It is clear that teams 13, 14, 15, 17 and 21 in general have a larger spread of results per instance, indicating their methods are either more susceptible to random events or possibly not yet converged after the time limit is reached. Team 1, 8, 11 and 20 generally have a low spread of the results; in particular team 20 has a spread of 0 for all instances because the submission always produces the same result.

The final ranking method, by mean rank over all instances, was decided as the scoring system at the onset of the competition. Any scoring system is biased in one way or another, e.g. the ranking transformation discards information on the relative performance among submissions on each instance. In order to investigate how the scoring system influenced the final results, different scoring systems were tested. In addition to ranking by mean rank, the following systems were compared:

- Ranking by mean: ranks according to the mean of the averaged results over all instances. This system is biased by the difference in magnitude of the objective function value between instances: larger instances will typically have a higher magnitude of objective function value, and thus will contribute more to the mean. Scoring well on large instances is thus important. However, the relative performance among submissions per instance is not lost.
- Ranking by geometric mean: ranks according to the geometric mean of the averaged results over all instances. This system is less biased by the difference in magnitude of the instance objective function value, and also maintains information on relative performance between submissions per instance.
- Ranking by total F1-score: this scoring system is adopted from Formula 1 racing point system (before 2010), and was used by the CHeSC 2011 competition (Burke et al. 2011). For each instance, the submissions are attributed points from 10 to 0. The highest average result is attributed 10 points, the second highest 8, the third 6, and finally 5, 4, 3, 2 and 1 to the remaining results in that order. The total number of points over all instances is then used to rank the submissions. Since the attribution of points is essentially a ranking transformation, relative performance between submissions per instance is lost. Furthermore, the method is biased towards the best results per instance.
- Ranking by mean best rank: this method differs from the competition ranking method by computing the ranks per instance based on the best result produced by each submission, rather than their averaged result. Therefore, this method considers the 'best' results for any submission, but may be biased by 'lucky results', and also loses information on relative performance between teams per instance. A summary of the best results found for all instances and teams can be found in Table 5.
- Ranking by mean rank over all runs: this method attributes a rank per instance to each run of all teams (i.e. a rank between 1 and 90 as there were 9 fully feasible teams and 10 runs with different random seed per team). Next, a mean rank is computed per team by computing the average rank over all instances and all runs of this team.

These 6 different scoring systems produce the final ranks[4] shown in Table 4. It is clear that the order of the best four submissions (Teams 1, 8, 11, and 20) is stable to the considered scoring mechanisms. However, there are some changes between positions 5 through 8, differentiating the teams ranking 5th according to the mean rank. It becomes apparent that team 17 produced better results than the other two teams at rank 5.

4.2 Properties of submitted algorithms

A variety of methods have been submitted to the MISTA 2013 challenge.

Toffolo et al. (2013) (Team 1) use a decomposition-based matheuristic. The approach considers resolution of several integer programming (IP) models, in three different stages. A feasible set of execution modes for the jobs is obtained, in the first stage. The second stage decomposes the problem into time windows. Each time window is solved with an IP model, considering the execution modes obtained from the first stage. The result of the second stage is a feasible solution, which is improved in the final stage by a hybrid local search based on forward-backward improvement (Lova et al. 2009; Valls et al. 2005) and an IP model to perform controlled changes to the solution.

[4] Details on how all these ranks were calculated can be reviewed in the detailed Google Docs Sheet, containing all results, at https://docs.google.com/spreadsheet/ccc?key=0Ar3CEQ-QxKb6dHZqbG9xNC0tUlN5UV95aGRsZFYyZHcq&usp=sharing.

Table 2 Average results per instance for submissions to the MISTA competition

Instance	Best found (Team)	Team 1	Team 2	Team 8	Team 11	Team 13	Team 14	Team 15	Team 17	Team 18	Team 20	Team 21
B-1	34900127 (11)	36650131.6	46260136.0	35950130.2	**35180128.0**	38390132.2	45250138.4	39780131.1	45450134.2	36850127.3	43200128.0	44550144.9
B-2	43400160 (1)	**44480161.5**	59610160.9	52600176.7	45430168.1	77520169.2	56860172.8	80820172.9	60440167.5	–	52600153.0	53420171.0
B-3	54500210 (11)	66520208.1	72860213	61880216.5	**55400211.0**	71200220.5	73530226.1	76420222.8	73530221.3	64740209.5	63800205	74720228.5
B-4	127400289 (8)	157360300.5	–	131650286.8	**130540284.4**	182260293.7	167040304.0	177100293.1	169560310.8	160970285.3	146900297.0	177770327.3
B-5	82000254 (11)	93090254.7	123160257.3	88900262.7	**83260254.3**	113860266.9	110460282	111100261.2	116500268.7	103860256.6	107500249.0	128060305.2
B-6	91200227 (11)	115080233.6	113690235.1	110370246.9	**95270231.7**	129010257.8	145570288.7	136300267.9	132000247.0	115480233.8	108300217.0	128680266.6
B-7	79200228 (11)	93140244.1	–	85130236.0	**80110232.1**	102790242.5	113410262.2	99820237.6	112880268.6	92580232.3	90500231.0	140970289.7
B-8	317600533 (11)	335130533.7	634320579.2	373480573.6	**331440547.7**	479160566.1	430380588.4	518640593.1	392940583.4	489190549.3	366200528.0	390210610.2
B-9	419200746 (11)	543180771.0	–	479350807.2	**426350755.4**	809010849.1	912471167.6	727550817.3	603760854.2	620010783.2	546500746	750791055.6
B-10	324900456 (11)	340420439.5	408050473.9	368230476.8	**333780459.6**	459970482.2	382340507.2	483260491.5	436690486.2	421020452.8	403430427.0	362130469.4
X-1	39200142 (1)	**40290143.9**	70990149.9	41710143.7	40490142.4	61810148.5	49850154.8	54170151.1	50270148.9	43920148.1	47800144.0	50950160.9
X-2	34900163 (11)	43740169.8	45180164.9	38280168.4	**35700164.1**	43330169.3	45150175.1	45150167.5	52000179.0	41530164.3	42300159.0	66740203.3
X-3	32400192 (11)	33940189.5	41560187.4	39240195.5	**33040193.2**	43910192.6	48310205.5	45520187.9	46030196.0	41130183.1	39100186.0	51910205.9
X-4	95500213 (11)	101310208.0	134630211.2	101080212.7	**97060212.3**	116900219.6	120890220.3	123080218.9	108620210.5	102820207.5	105400198.0	121490229.5
X-5	176800374 (11)	208560378.8	272290390.6	196940389.7	**178450373.0**	232370397.4	240170430.6	230450380.8	245070401.9	221130378.4	207650367.0	268910482.6
X-6	71900232 (11)	93560246.8	104840244.9	89310255.1	**73840240.9**	137740266.4	107710284.3	111760248.8	123840278.3	95380231.0	87250214.0	178060328.1
X-7	86100237 (11)	97010236.7	140710242.2	89690236.6	**86750235.5**	118030243.2	123960253.3	115720241.1	126470268.6	98130231.3	99300229.0	135930286.3
X-8	123300283 (11)	150420291.7	205300295.6	146040305.8	**125720289.3**	184310313.8	185130356.3	183070302.3	173170312.6	148130286.7	166050270.0	264200376.3
X-9	326800643 (11)	422520660.0	672350739.4	378050691.9	**330320647.1**	592750739.6	670700995.6	516180699.4	528240754.0	492920681.7	513000635.0	620020908.0
X-10	160000381 (11)	187040385.5	–	174720399.5	**161440382.3**	260000414.0	207090441.0	241140400.2	259690429.1	219410395.4	197400376.0	378300597.7
Mean	136060304.5	160172309.5	–	154130320.6	**138978807.6**	212716329.2	211813872.7	205851824.3	192857836.0	–	171709298.0	219390882.4
Geom. mean	98920555.1	114261313.6	–	111077244.1	**101043319.3**	144307583.4	140801578.2	142032106.4	139086621.6	–	121054656.9	155392199.6
F1-score		126	–	142	**196**	45	43	45	45	–	110	28

Results are averaged over 10 runs with different seeds for each instance. Bold highlighting indicates the best average result per instance. Dashes indicate infeasible results generated for at least one run

Table 3 Ranking per instance based on the averaged result per submission of the final phase

Instance	Team 1	Team 8	Team 11	Team 13	Team 14	Team 15	Team 17	Team 20	Team 21
B-1	3	2	**1**	4	8	5	9	6	7
B-2	**1**	4	2	8	6	9	7	3	5
B-3	4	2	**1**	5	7	9	6	3	8
B-4	4	2	**1**	9	5	7	6	3	8
B-5	3	2	**1**	7	5	6	8	4	9
B-6	4	3	**1**	6	9	8	7	2	5
B-7	4	2	**1**	6	8	5	7	3	9
B-8	2	4	**1**	8	7	9	6	3	5
B-9	3	2	**1**	8	9	6	5	4	7
B-10	2	4	**1**	8	5	9	7	6	3
X-1	**1**	3	2	9	5	8	6	4	7
X-2	5	2	**1**	4	7	6	8	3	9
X-3	2	4	**1**	5	8	6	7	3	9
X-4	3	2	**1**	6	7	9	5	4	8
X-5	4	2	**1**	6	7	5	8	3	9
X-6	4	3	**1**	8	5	6	7	2	9
X-7	3	2	**1**	6	7	5	8	4	9
X-8	3	2	**1**	7	8	6	5	4	9
X-9	3	2	**1**	7	9	5	6	4	8
X-10	3	2	**1**	8	5	6	7	4	9
Average rank	3.05	2.55	**1.10**	6.75	6.85	6.75	6.75	3.60	7.60
Final rank	3	2	**1**	5	8	5	5	4	9

Ties are resolved by the average rank. Bold highlighting indicates best rank. Note that team 2 and team 18 are not ranked due to infeasible results

Table 4 Ranking results of final submissions by different ranking methods

Ranking method.	Team								
	1	8	11	13	14	15	17	20	21
Mean	3	2	**1**	8	7	6	5	4	9
Geom. Mean	3	2	**1**	8	6	7	5	4	9
Mean rank[a]	3	2	**1**	6	8	6	6	4	9
F1 score	3	2	**1**	6	8	6	6	4	9
Best rank	3	2	**1**	7	5.5	5.5	8	4	9
Mean over all runs rank	3	2	**1**	8	6	7	5	4	9

Considered methods: ranking by mean over all instances, ranking by geometric mean over all instances, ranking by mean rank over all instances, ranking by F1-score, ranking by mean best rank, ranking by mean over all runs rank. Bold highlights indicate best ranked

[a] Competition ranking method

Geiger (2013) (Team 8) proposes an iterated variable neighbourhood search with four different neighbourhoods: exchange, inversion, single mode change and double mode change. Random initial solutions are generated with an additional mode repair procedure. A serial schedule generation scheme (SSGS) constructs schedules with a given activity list and mode assignment.

Asta et al. (2013) (Team 11) apply a combination of Monte-Carlo Tree Search (MCTS) and hyper-heuristics. MCTS first partitions the projects from which their ordering is then determined. An initial solution is constructed by randomly scheduling the activities in each project, while only taking into account the precedence constraints. A multi-threaded hyper-heuristic with threshold acceptance is used

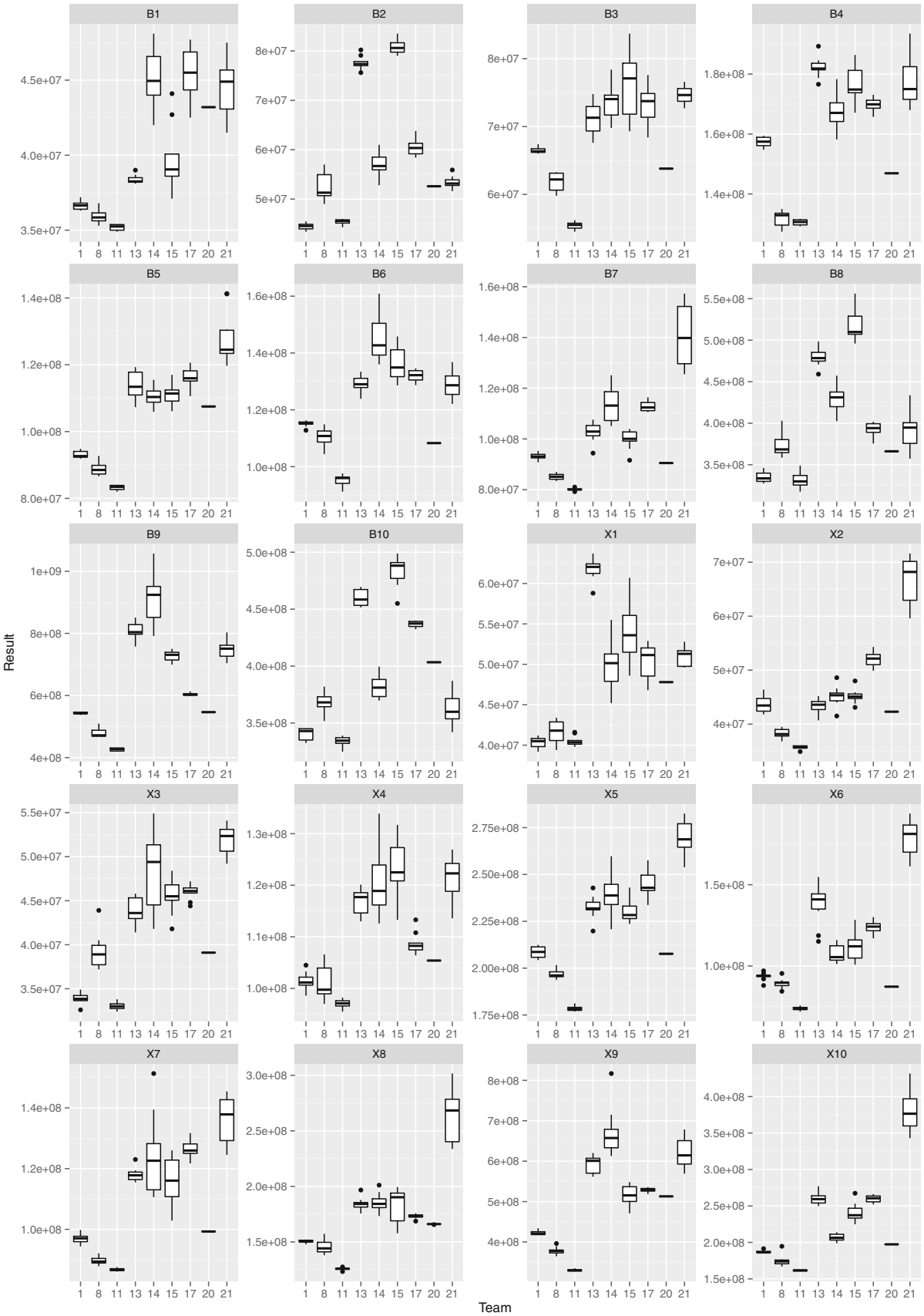

Fig. 3 Summarizing boxplots of participants' results per instance

Table 5 Best results per instance and per team for submissions to the MISTA competition

Instance	Best found (Team)	Team 1	Team 2	Team 8	Team 11	Team 13	Team 14	Team 15	Team 17	Team 18	Team 20	Team 21
B-1	34900127 (11)	36300132	43800133	35300125	**34900127**	38100133	42000139	37100127	42500133	36100130	43200128	41500144
B-2	43400160 (1)	**43400160**	55400164	49000176	44300167	75600165	52800170	79000174	58400160	53700161	52600153	51600168
B-3	54500210 (11)	66000207	69400218	59800215	**54500210**	67600221	69800225	69300214	68400225	62100211	63800205	72700222
B-4	127400289 (8)	154800295	198600300	**127400289**	129200287	176600294	158200308	167100287	165700308	156800283	146900297	168000315
B-5	82000254 (11)	91900254	105400255	86600254	**82000254**	107300263	105900299	106100255	110600270	100000254	107500249	119700305
B-6	91200227 (11)	112800232	109700235	104400242	**91200227**	123900257	136000309	128600259	128700248	111900230	108300217	122100253
B-7	79200228 (11)	90800246	134800242	83400234	**79200228**	94400239	105100254	91600228	110600270	90300233	90500231	125500267
B-8	317600533 (11)	327600529	601000566	358500568	**317600533**	459100544	402500549	495900584	375500578	472800533	366200528	357400588
B-9	419200746 (11)	537300769	787500851	467400796	**419200746**	757700837	791401176	699600812	597900851	613000779	546500746	704201024
B-10	324900456 (11)	332500447	385000472	351800469	**324900456**	451600472	369700510	455000483	432400487	413300452	403300427	342000474
X-1	39200142 (1)	**39200142**	63700150	39400142	39800142	58800152	45200148	48600150	46800149	43200149	47800144	49600156
X-2	34900163 (11)	41800165	40200163	36800165	**34900163**	40700170	41500166	43100162	49900177	40100163	42300159	59600189
X-3	32400192 (11)	32600188	39400187	37200195	**32400192**	41400189	41800202	41800182	44400195	38800183	39100186	49200204
X-4	95500213 (11)	98600207	122600211	97000215	**95500213**	113000213	112600211	113300210	106400208	98800204	105400198	113600235
X-5	176800374 (11)	204300375	252800389	193800386	**176800374**	219800378	220800423	223600376	233700406	217600373	207600367	253900464
X-6	71900232 (11)	88000240	93400240	84400253	**71900232**	115100253	101200282	100800245	117000269	91900226	87200214	161300317
X-7	86100237 (11)	94400234	132700242	87900231	**86100237**	115500238	110600245	102900235	121700271	95300230	99300229	124500276
X-8	123300283 (11)	147800289	189100299	138000296	**123300283**	175600308	173300357	157800298	168700311	143500282	165600270	233700358
X-9	326800643 (11)	416900662	648600741	364500688	**326800643**	561400721	612900970	470800696	518500747	456200682	513000635	569200866
X-10	160000381 (11)	185100385	261100394	166900402	**160000381**	249900412	198900431	224800393	252100418	205000396	197400376	343000586

Bold highlighting indicates the best result per instance. Note that the table includes best results for teams 2 and 18, as they did produce results for all instances (but some infeasible). They are however not considered for any ranking method

Table 6 Properties of the nine algorithms in the competition final

Team	Search strategy	NN	Population	SGS	FB	Solver	Parallel
1	ILS/LNS	1	No	Serial	Yes	Gurobi 5.5	Yes
8	VNS/ILS	4	No	Serial	No	–	Yes
11	HH	13	Yes	Serial	No	–	Yes
13	FILS	2	No	Serial	No	–	–
14	EA + ILS	3	Yes	Serial	Yes	–	–
15	SA + TS	2–3	No	?	No	–	–
17	ILS	2	No	Serial	Yes	–	Yes
20	LNS	1	No	None	No	ILOG Cplex/CPOptimizer	–
21	SD	4	No	Serial	No	SCIP	–

to control 13 neighbourhood moves, which change mode assignments, activity ordering, and project ordering. Furthermore, hill climbing heuristics are used within the hyper-heuristic to intensify the search, similar to the local search part of memetic algorithms.

Borba et al. (2013) (Team 13) propose a stochastic local search procedure with two neighbourhoods (single mode change, swap). Dynamic programming determines the initial mode selection.

Bouly et al. (2013) (Team 14) make use of an evolutionary algorithm and local search. A triplet encoding including the sequence of jobs and the chosen modes combined with a SSGS and a repair procedure to produce feasible schedules. The local search applies three different neighbourhoods and is used as a post-processing procedure for the evolutionary algorithm.

Alonso-Pecina et al. (2013) (Team 15) propose a three-phase approach. First, feasible solutions are constructed for each project separately, which are afterwards combined into an initial feasible solution for the complete problem. The second phase improves the solution with a simulated annealing metaheuristic. It applies two neighbourhoods that change mode assignments and activity ordering. The solution is further improved in the third phase with a tabu search algorithm exploring the two previous neighbourhoods, and one additional neighbourhood that changes two mode assignments simultaneously.

López-Ibáñez et al. (2013) (Team 17) used an automatic design method to develop an algorithm for the MRCMPSP. The design method uses 'irace' to find the best combination of algorithm parameters out of a set. The proposed algorithm is an iterated local search with two neighbourhoods (single mode change, swap). Several initial solutions are generated using different construction heuristics.

Artigues and Hebrard (2013) (Team 20) introduce a hybrid approach, which focuses mainly on mode selection. It includes a MIP relaxation model for producing an initial mode assignment. The improvement method alternates between mode selection, conducted with large neighbourhood search, and fixed-mode optimization based on a standard Constraint Programming model.

Schnell extends SCIP (Schnell 2013) (Team 21) with a multi-mode cumulative constraint, enabling generation of optimal solutions for 2-project instances in less than 35 sec. For the larger instances, a steepest descent local search heuristic with four neighbourhoods is applied. An initial solution is constructed by solving all projects separately for minimal makespan, and sequencing the projects for a minimal total project delay. Sorting all activities in this sequence by increasing starting times results in the initial activity list to which a SSGS is applied. Two neighbourhoods relax up to two projects, enabling generation of a different solution. One randomly selects the projects, while the other selects projects based on the lower bound of the relaxed solution. A third neighbourhood relaxes some of the activities that are scheduled at the end of the current solution. A fourth relaxes the mode assignments in an effort to minimize the complete renewable resource energy. The resulting partially relaxed problem is solved with SCIP.

Table 6 shows the presence of different components in the nine feasible submissions. The following components and properties are shown:

- Search strategy: The reported search strategy: iterated local search (ILS), variable neighbourhood search (VNS), hyper-heuristic (HH), first improving local search (FILS), evolutionary algorithm (EA), simulated annealing (SA), tabu search (TS), large neighbourhood search (LNS) and steepest descent (SD).
- NN: number of neighbourhoods.
- Population: whether or not the approach is population based.
- SGS: if and which schedule generation scheme is used.
- FB: whether or not the forward-backward or double justification procedure is used (Lova et al. 2009; Valls et al. 2005).

– Solver: if and which (general purpose) solver is used.
– Parallel: parallel execution.

All teams submitted a heuristic procedure in which some included exact components. Some model elements are common to most approaches, namely an activity list and a mode assignment representation combined with a SSGS. The single mode change neighbourhood, where the mode assignment of a single job is changed, is part of most teams' approaches. It is noteworthy that every team has a different approach to generate feasible initial solution(s), and especially to find a feasible mode assignment, e.g. random, monte-carlo tree search, multi-dimensional knapsack,. . . The teams do not all explicitly apply parallel execution. Some teams use a CP or MIP for solving subproblems to optimality or for conducting a large neighbourhood search.

5 Conclusion

This paper introduced the MRCMPSP, subject of the MISTA 2013 challenge. The submitted approaches and the results have been thoroughly analysed and discussed. The problem shows to be very challenging, and a large variety of algorithms and results can be observed. The challenge serves as a new benchmark platform for the MRCMPSP and stimulates research for this and related problems. Many instances and results are available for comparison and facilitate future research on the problem.

Future research can take advantage of the collected knowledge, for example by (1) exploring synergy by combining the key components of the feasible algorithms, (2) applying the collected set of algorithms in a portfolio approach.

Acknowledgments We would like to thank the challenge sponsors: iMinds, CONUNDRA and OM Partners. We would also like to thank all participating teams. This research is partially supported by a Ph. D. Grant of the agency for Innovation by Science and Technology (IWT).

Appendix 1: File formats

Instance input file format

All problem inputs are integer valued, and comply with the following file format. Each problem instance is defined by $n + 1$ files, one global file and n project files.

Global file

The global file contains the number of projects, the release dates, the path to the project files, and the global resource capacities. Values are space separated and should respect the following order:

```
Number of projects
Release date project 0
Critical path duration project 0
Path to project file of project 0
Release date project 1
Critical path duration project 1
Path to project file of project 1
...
Release date project n-1
Critical path duration project n
   -1
Path to project file of project n
   -1
Number of resource types
Global resource capacities (-1 is
    not global)
```

Project files

Each project is defined in a separate file. The relative path to this file is given in the global file (e.g. *j20.mm/j2010_1.mm*). A project file contains the number of activities (including the dummy activities), the precedence relations between the activities, the execution modes, and the local resource capacities. The individual projects are represented in the PSPLIB[5] MRCPSP file format.

Important note: In order to preserve compatibility with PSPLIB, the global resources always overwrite the local resources and their capacities. For example, consider the following global resource capacities (16,−1,−1,−1) and local resource capacities (RR1 = 14, RR2 = 18, NR1 = 60, NR2 = 68), then there exists one global renewable resource with capacity 16, one local renewable resource with capacity 18 and two non-renewable resources with capacities 60 and 68. The local renewable resource with capacity 14 can be ignored.

Solution output format

A solution for the MRCMPSP is defined as follows:
For each activity j of project i,

– the selected mode m_{ij},
– and the start time s_{ij},

must be given.
An example is given in Table 7.

[5] PSPLIB benchmark website: http://www.om-db.wi.tum.de/psplib/main.html.

Table 7 Solution output format example

Project i	Activity j	Mode m_{ij}	Start time s_{ij}
0	0	2	0
0	1	1	5
...	...	...	...
1	0	0	4
1	1	0	8
...	...	...	...

Solution validator

The validator checks feasibility of a solution, and computes the total objective cost. It is available for download at the MISTA challenge website (Wauters et al. 2013). The validator requires Java. It can be run as follows:

```
java -jar MRCMPSP-validator.jar
   global_problem_file
   schedule_file
```

The validator was used for evaluating the quality of solutions produced by the algorithms submitted to the MISTA 2013 challenge. Note that the validator does not report validity of the instances.

Example

An example MRCMPSP instance and a corresponding feasible solution can be found on the challenge website. The instance has 2 projects, 10 jobs per project and 1 global renewable resource with capacity 12. The global file corresponding to this example is

```
2
0
17
j10.mm/j1010_1.mm
6
22
j10.mm/j1010_2.mm
4
12  -1  -1  -1
```

References

Alonso-Pecina, F., Pecero, J. E., & Romero, D. (2013). A three-phases based algorithm for the multi-mode resource-constrained multi-project scheduling problem. In G. Kendall, G. Vanden Berghe, & B. McCollum (Eds.), *Proceedings of the 6th Multidisciplinary International Scheduling Conference* (pp. 812–814). : MISTA.

Artigues, C., & Hebrard, E. (2013). MIP relaxation and large neighborhood search for a multi-mode resource-constrained multi-project scheduling problem. In G. Kendall, G. Vanden Berghe, & B. McCollum (Eds.), *Proceedings of the 6th Multidisciplinary International Scheduling Conference* (pp. 815–819). : MISTA.

Asta, S., Karapetyan, D., Kheiri, A., Özcan, E., & Parkes, A. (2013). Combining monte-carlo and hyper-heuristic methods for the multi-mode resource-constrained multi-project scheduling problem. In G. Kendall, G. Vanden Berghe, & B. McCollum (Eds.), *Proceedings of the 6th Multidisciplinary International Scheduling Conference* (pp. 836–839). : MISTA.

Blazewicz, J., Lenstra, J., & Rinnooy Kan, A. (1983). Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics*, *5*(1), 11–24. doi:10.1016/0166-218X(83)90012-4.

Borba, L.M., Benavides, A.J., Zubaran, T., Carniel, G.C., & Ritt, M. (2013). A simple stochastic local search for multi-mode resource-constrained multi-project scheduling. In G. Kendall, G. Vanden Berghe, & B. McCollum (Eds.), *Proceedings of the 6th Multidisciplinary International Scheduling Conference* (pp. 826–830). : MISTA.

Bouly, H., Dang, D.C., Moukrim, A., & Xu, H. (2013). Evolutionary algorithm and post-processing to minimize the total delay in multi-project scheduling. In G. Kendall, G. Vanden Berghe, & B. McCollum (Eds.), *Proceedings of the 6th Multidisciplinary International Scheduling Conference* (pp. 831–835). : MISTA.

Burke, E. K., Gendreau, M., Hyde, M., Kendall, G., McCollum, B., Ochoa, G., et al. (2011). The cross-domain heuristic search challenge - an international research competition. In C. A. Coello Coello (Ed.), *Learning and Intelligent Optimization, Lecture Notes in Computer Science* (Vol. 6683, pp. 631–634). Berlin Heidelberg: Springer. doi:10.1007/978-3-642-25566-3_49.

Geiger, M.J. (2013). Iterated variable neighborhood search for the resource constrained multi-mode multi-project scheduling problem. In G. Kendall, G. Vanden Berghe, & B. McCollum (Eds.), *Proceedings of the 6th Multidisciplinary International Scheduling Conference* (pp. 807–811). : MISTA.

Haspeslagh, S., De Causmaecker, P., Schaerf, A., & Stølevik, M., (2012). The first international nurse rostering competition 2010. *Annals of Operations Research*, pp *1–16*, doi:10.1007/s10479-012-1062-0.

Kolisch, R., & Drexl, A. (1997). Local search for nonpreemptive multi-mode resource-constrained project scheduling. *IIE Transactions*, *29*, 987–999.

López-Ibáñez, M., Mascia, F., Marmion, M.E., & Stützle, T. (2013). Automatic design of a hybrid iterated local search for the multi-mode resource-constrained multi-project scheduling problem. In G. Kendall, G. Vanden Berghe, & B. McCollum (Eds.), *Proceedings of the 6th Multidisciplinary International Scheduling Conference* (pp. 820–825). : MISTA.

Lova, A., Tormos, P., Cervantes, M., & Barber, F. (2009). An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes. *International Journal of Production Economics*, *117*(2), 302–316. doi:10.1016/j.ijpe.2008.11.002.

McCollum, B., Mcmullan, P., Burke, E. K., Parkes, A. J., & Qu, R. (2007). The second international timetabling competition: Examination timetabling track.

Post, G., Di Gaspero, L., Kingston, J., McCollum, B., Schaerf, A. (2013). The third international timetabling competition. *Annals of Operations Research* pp 1–7, doi:10.1007/s10479-013-1340-5, http://dx.doi.org/10.1007/s10479-013-1340-5

Schnell, A. (2013). A hybrid local search algorithm integrating an exact cp-sat approach to solve the multi-mode resource-constrained multi-project scheduling problem. In G. Kendall, G. Vanden Berghe, & B. McCollum (Eds.), *Proceedings of the 6th Multi-*

disciplinary International Scheduling Conference (pp. 802–806). : MISTA.

Solnon, C., Cung, V. D., Nguyen, A., & Artigues, C. (2008). The car sequencing problem: Overview of state-of-the-art methods and industrial case-study of the roadef2005 challenge problem. *European Journal of Operational Research*, *191*(3), 912–927. doi:10.1016/j.ejor.2007.04.033.

Toffolo, T. A. M., Santos, H. G., Carvalho, M., & Soares, J. (2013). An integer programming approach for the multi-mode resource-constrained multi-project scheduling problem. In G. Kendall, G. Vanden Berghe, & B. McCollum (Eds.), *Proceedings of the 6th Multidisciplinary International Scheduling Conference* (pp. 840–847). : MISTA.

Valls, V., Ballestin, F., & Quintanilla, S. (2005). Justification and rcpsp: A technique that pays. *European Journal of Operational Research*, *165*(2), 375–386. doi:10.1016/j.ejor.2004.04.008.

Wauters, T., Verbeeck, K., Verstraete, P., & De Causmaecker, P. (2012). Real-world production scheduling for the food industry: An integrated approach. *Engineering Applications of Artificial Intelligence*, *25*(2), 222–228. doi:10.1016/j.engappai.2011.05.002.

Wauters, T., Kinable, J., Smet, P., Vancroonenburg, W., Verstichel (2013). The MISTA 2013 challenge. http://allserv.kahosl.be/mista2013challenge/, [Online]