

A combinatorial approach to multi-skill workforce scheduling

Citation for published version (APA):

Firat, M., & Hurkens, C. A. J. (2010). *A combinatorial approach to multi-skill workforce scheduling*. (BETA publicatie : working papers; Vol. 327). Technische Universiteit Eindhoven.

Document status and date:

Published: 01/01/2010

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

A combinatorial approach to multi-skill workforce scheduling

Murat Firat, Cor Hurkens

Beta Working Paper series 327

BETA publicatie	WP 327 (working paper)
ISBN	978-90-386-2362-7
ISSN	
NUR	982
Eindhoven	September 2010

Editorial Manager(tm) for Journal of Scheduling
Manuscript Draft

Manuscript Number: JOSH678

Title: A combinatorial approach to multi-skill workforce scheduling

Article Type: Paper

Keywords: project scheduling; multi-skill workforce scheduling; mixed integer programming

Corresponding Author: Mr. Murat Firat, M.S.

Corresponding Author's Institution: Eindhoven University of Technology

First Author: Murat Firat, M.S.

Order of Authors: Murat Firat, M.S.; Cor Hurkens, Ph.D.

A combinatorial approach to multi-skill workforce scheduling

Murat Firat ^{†*} Cor Hurkens [†]

Abstract

This paper deals with scheduling complex tasks with an inhomogeneous set of resources. The problem is to assign technicians to tasks with multi-level skill requirements. Here the requirements are merely the presence of a set of technicians that possess the necessary capabilities. An additional complication is that a set of combined technicians stays together for the duration of a work day. This typically applies to scheduling of maintenance and installation operations. We build schedules by repeated application of a flexible matching model that selects tasks to be processed and forms groups of technicians assigned to combinations of tasks. The underlying mixed integer programming (MIP) model is capable of revising technician-task allocations and performs very well, especially in the case of rare skills.

Keywords: project scheduling, multi-skill workforce scheduling, mixed integer programming.

1 Introduction

As specialization in production and maintenance increases, the importance of skill management in employee scheduling grows significantly. Especially when activities require skills from several specialization fields at different levels, skill management becomes more challenging. Multi-skilled employee scheduling can be encountered for example in companies having operations like maintenance, construction and installation in which the work is carried at different physical locations. Then it makes sense to keep a combined group of workers together for a workday.

We developed an approach to this problem based on a hybrid combination of MIP models and applied it on maintenance instances provided by France Telecom. Since the problem instances of France Telecom have been used as a test bed for a computational challenge held in 2007, we can compare the performance of our method to other approaches tackling the same problem.

In each of the problem instances, we are given a set of tasks to be performed. For each workday, tasks are combined for being performed by assigned technicians without interruption. Among tasks, there may be precedence relations requiring that a certain task p must be completed before another task q can be processed. In this case we say p precedes q or q succeeds p . A task can be processed by a set of technicians provided that the collective capabilities of this group are above a certain threshold. The capabilities or skills required by tasks and supplied by technicians, are described in terms of *domains* and *levels*.

*Corresponding author: m.firat@tue.nl. Supported by France Telecom/TUE Research agreement No. 46145963.

[†]Department of Mathematics and Computer Science, TU Eindhoven, P.O. Box 513, 5600 MB Eindhoven, The Netherlands.

1
2
3
4
5
6
7
8 Some tasks can be outsourced by hiring external companies. An outsourced task can be
9 discarded and need not be scheduled. If a task is outsourced, then so must the ones succeeding
10 it. Each task has an outsourcing cost and outsourcing is limited by a budget.

11 Besides the tasks, we are given a list of technicians with their availabilities on each workday.
12 Technicians grouped on a certain workday to carry out a task must stay together for the duration
13 of a workday. The group may perform different tasks but one at a time. Any travel or setup
14 time in between tasks is disregarded.

15 The value of a schedule is measured by the makespan and a weighted sum of completion
16 times. For this purpose tasks are partitioned in different priority classes. The latest completion
17 time of tasks under a priority class is considered and the weighted sum of these completion times
18 is taken. Evidently the most urgent tasks have highest weights. All tasks within a priority class
19 have the same weight.

20 The proposed combinatorial approach is composed of three parts: *preprocessing*, *finding*
21 *lower bounds* and *constructing schedules*. In the preprocessing part we calculate several charac-
22 teristic properties of tasks. In the second part, lower bounds of the schedule costs are calculated
23 by solving a simplified problem in which skill requirements and precedence relations of tasks are
24 relaxed and preemption is allowed. The problem of calculating lower bounds and deciding which
25 tasks to outsource is formulated as a MIP model. This model assumes that priority classes are
26 handled sequentially. We run the lower bound model for all possible priority permutations. We
27 follow the idea that smaller lower bound values promise low-cost schedules.

28 We construct schedules by finding simultaneous technician-task assignments and define a bi-
29 partite graph in which many-to-one type matchings correspond to technician-task assignments.
30 Matchings are restricted by skill requirements, precedence relations and total durations. The
31 problem of finding matchings is formulated as a MIP model. The proposed algorithm starts the
32 construction of a day schedule with an initial matching phase and proceeds to the second phase
33 in which the partially constructed schedule is extended by adding more tasks and reallocat-
34 ing technicians flexibly. The idea of finding simultaneous technician-task assignments was first
35 introduced by Hurkens (2009) in the 2007 ROADEF Challenge and this approach was ranked
36 first in the final stage among 11 qualified participants. In this paper we improved the idea of
37 Hurkens (2009) by introducing several measurements for tasks and a more flexible MIP model
38 for constructing day schedules.

39 The paper is organized as follows. In Section 2 we give a detailed description of the problem
40 under consideration. Section 3 gives an overview of the studies in the literature related to our
41 problem. Our solution approach is explained in details in Section 4 and computational results
42 are presented in Section 5. Conclusions and further discussion are included in Section 6.
43
44

45 2 Problem description and notation

46 2.1 Tasks

47 The problem under consideration has been defined by France Telecom in the 2007 ROADEF
48 Challenge (Dutot et al. 2006). In the problem, a set $J = \{1, \dots, n\}$ of tasks is given. Each task
49 has a fixed duration not longer than a workday. For a task $j \in J$, let d_j denote its duration. We
50 have $d_j \in \{1, 2, \dots, H\}$ where $H = 120$ is the number of time units in a workday. Precedence
51 relations of task j enforce that all tasks in $Pred(j)$ must be completed before the task j starts.
52
53
54
55
56
57
58
59
60
61
62

1
2
3
4
5
6
7
8 A task $k \in \text{Pred}(j)$ is said to be a *predecessor* of task j and their precedence relation is denoted
9 by $k \rightarrow j$. It is also said that task j is a *successor* of task k . Moreover the urgency of task j is
10 represented by $P(j) \in \{1, 2, 3, 4\}$ denoting its priority class.

11 Tasks require certain skill qualifications. The requirements are specified as the number
12 of technicians needed for each level of specialization. We will use the term “*skill domain*”
13 for a specialization field in the rest of the paper. The degree of expertise is represented by
14 levels in each skill domain. The skills required to perform a task are expressed by a skill
15 requirement matrix. Let task j have skill requirement matrix $RQ_j \in \mathbb{R}^{\mathbb{L} \times \mathbb{S}}$ where $\mathbb{S} (\mathbb{L})$ is the
16 set of skill domains (skill levels). Entries of RQ_j specify how many technicians are needed for
17 the corresponding skill domain-skill level pair. For example, $RQ_j^{(2,1)} = 2$ means that task j
18 requires (at least) 2 technicians that have a competence in domain 1 of level 2 or higher. Skill
19 requirements are cumulative in the sense that any requirement in higher level is carried to lower
20 levels. Therefore we have:
21
22

$$23 \quad l' \leq l \Rightarrow RQ_j^{(l',s)} \geq RQ_j^{(l,s)}, \quad \forall j \in J, l, l' \in \mathbb{L}, s \in \mathbb{S}$$

24
25
26 As mentioned in the introduction, external companies can be hired to perform some tasks.
27 The cost of outsourcing a task j is denoted by c_j .
28
29

30 2.2 Technicians

31
32 We are given a set $T = \{1, \dots, m\}$ of technicians to perform the tasks. Technicians have
33 availability periods specified as follows:
34

$$35 \quad A(t, d) = \begin{cases} 1 & \text{if technician } t \text{ is available on day } d, \\ 0 & \text{otherwise.} \end{cases}$$

36
37
38 Skills of technicians are expressed with skill vectors and skill matrices. Entries of a skill
39 vector specify the technician’s level of expertise in skill domains. Skill matrix entries show
40 whether a technician is qualified for a particular skill domain-skill level pair. If a technician is
41 qualified at a skill level, then that technician is also qualified at lower levels in the same skill
42 domain. Let technician t have skill vector $SV_t \in \{0, 1, \dots, |\mathbb{L}|\}^{\mathbb{S}}$ and skill matrix $SM_t \in \{0, 1\}^{\mathbb{L} \times \mathbb{S}}$.
43 The skill level of t in skill domain s is given by $SV_t^{(s)}$ and for the skill matrix we have the
44 following:
45
46

$$47 \quad SM_t^{(l,s)} = \begin{cases} 1 & \text{if } l \leq SV_t^{(s)}, \\ 0 & \text{otherwise.} \end{cases}$$

48
49
50 For example, in a problem instance with $|\mathbb{L}| = |\mathbb{S}| = 3$, a technician having the skill vector
51 $(3, 0, 2)$ has the following skill matrix:
52
53

$$54 \quad \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

2.3 The concept of a workday

In our problem, the time axis is partitioned into intervals of length H . These successive intervals represent workdays. Within a certain time interval (workday), the technicians performing a certain task must work together during this interval. Another important restriction is that the processing of each task must stay within a time interval.

2.4 Properties of schedules

A schedule defines the set of tasks performed, the time at which tasks are completed and the assignments of technicians to these tasks. The technician-task assignment is represented by defining a set of teams for each workday. A team has a number of technicians to perform the assigned tasks. Let τ denote a team. Then the set $T(\tau)$ denotes its technicians and $J(\tau)$ denotes the set of tasks assigned to τ . In a feasible schedule the following conditions must hold for each team τ :

$$\begin{aligned} \sum_{t \in T(\tau)} SM_t^{(l,s)} &\geq RQ_j^{(l,s)}, \quad \forall (l,s) \in \mathbb{L} \times \mathbb{S}, \quad \forall j \in J(\tau) \\ \sum_{j \in J(\tau)} d_j &\leq H \end{aligned}$$

No preemption is allowed while performing a task. Let CT_j be the completion time of task j . Precedence relations enforce the following:

$$CT_j - d_j \geq CT_k, \quad \forall k \in Pred(j), \quad \forall j \in J$$

There is budget B for outsourcing tasks. The total cost of outsourced tasks must not exceed B . Let Ω denote the set of outsourced tasks. We follow the convention that the set $\{k \in J \mid \Omega \cap Pred(k) \neq \emptyset\} \subseteq \Omega$, that is, successors of outsourced tasks are outsourced as well.

The makespan of a priority class p is defined by $C_p = \max\{CT_j \mid P(j) = p, j \notin \Omega\}$. The overall makespan is the length of the schedule and denoted by $C_0 = \max\{CT_j \mid j \in J \setminus \Omega\}$. Note that outsourced tasks do not contribute to the schedule cost. Priority class 0 is an artificial priority class that is used to include the overall makespan in the quality evaluation of schedules and of course every task belongs to priority class 0. The cost of a schedule is calculated by the weighted sum of priority makespans $\sum_p w(p)C_p$ and in the benchmark instances of France Telecom the weights are as follows:

p	0	1	2	3	4
$w(p)$	1	28	14	4	0

3 Literature review

The problem considered in this paper is a generalization of the *resource constraint project scheduling problem*, shortly RCPSP, with some additional aspects and it has been shown to be NP-Hard by Blazewicz et al. (1983). A basic review of the state of art in RCPSP is provided by Brucker et al. (1999). A survey of variants of RCPSP has been recently published by Hartmann and Briskorn (2009).

1
2
3
4
5
6
7
8 **3.1 Use of skills**

9
10 In our problem the duration of a task does not vary with the skills of assigned technicians
11 and skills do not change as no learning or forgetting effect is considered, so skills are homoge-
12 neous and static in the terminology of Heimerl and Kolisch (2009). Li and Womer (2009) and
13 Bellenguez and Neron (2007) assume workers are “unary resources” which means that while
14 processing a task they can only work in one skill domain. In our problem we assume that tech-
15 nicians contribute simultaneously in all possible domains while processing a task. Let us call
16 this assumption on resources *simultaneous skill use*. Valls et al. (2009), Heimerl and Kolisch
17 (2009), Drezet and Billaut (2008) and Ballou and Tayi (1996) also make the *simultaneous skill*
18 *use* assumption.
19
20

21 **3.2 Interpretation of expertise in a skill domain**

22
23 The majority of studies encountered in the literature treat human resources as skilled or un-
24 skilled in skill domains. To the best of our knowledge, there are few studies that consider
25 skill (or expertise) levels of employees in a skill domain (Bellenguez and Neron 2007, Valls et
26 al. 2009). Some studies assume that employees have skill efficiencies (Gutjahr et al. 2008, Wu
27 and Sun 2006) or competence scores (Yoshimura et al. 2006, Heimerl and Kolisch 2009). The
28 drawback of scoring employee expertise rather than leveling is the difficulty in expressing skill
29 requirements of tasks clearly. In case of scoring the expertise, if a task requires an expert, then
30 its skill requirement is merely a high value in a domain. This high demand may be satisfied by
31 collecting many non-expert employees. On the other hand if expertise is expressed by levels in
32 a domain, then a task requiring an expert can only be satisfied by assigning the needed expert.
33 Gutjahr et al. (2008), using both skill efficiency and competence score for employees, limit the
34 number of employees that can be assigned to a task and define special variables for experts
35 to handle this issue in their nonlinear MIP model. The authors also suggest a transformation
36 function for mapping competence scores to skill efficiencies when employee competence are as-
37 sumed to be dynamic. Yoshimura et al. (2006) use a specific parameter for experts or so-called
38 “project leaders” in order to select an expert for each project. Ballou and Tayi (1996) use skill
39 efficiencies as cost coefficients representing the appropriateness of workers to tasks.
40
41
42
43
44

45 **3.3 Outsourcing**

46
47 Outsourcing is an option for some tasks to get performed by external resources. Heimerl and
48 Kolisch (2009), Wu and Sun (2006) and Bassett (2000) consider explicitly using external re-
49 sources and they minimize the cost occurring due to outsourcing. In our case, we have an
50 outsourcing budget and we try to maximize the number of outsourced tasks with the hope to
51 construct low-cost schedules.
52
53

54 **3.4 Solution approaches**

55
56 In multi-mode RCPS (or MM-RCPS), there are multiple ways (or modes) to perform activ-
57 ities (or tasks, jobs) and each mode is defined by a duration and certain amounts of resources
58 (Reyck and Herroelen 1999, Bellenguez and Neron 2007). In multi-skilled workforce scheduling
59 problem (or MSPSP), a combination of employees who can perform a task and the time they
60
61
62

1
2
3
4
5
6
7
8 need to perform that task corresponds to a mode in MM-RCPSP. The skills of employees define
9 the dedicated amount of resources. Depending on the assumptions, the duration of performing
10 a task by an employee combination may be fixed (Bellenguez and Neron 2007, Ballou and
11 Tayi 1996, Drezet and Billaut 2008) or may vary with skill efficiencies (Valls et al. 2009, Heimerl
12 and Kolisch 2009, Gutjahr et al. 2008, Wu and Sun 2006). However, as Drezet and Billaut
13 (2008), Heimerl and Kolisch (2009) and Bellenguez and Neron (2007) point out, it is not suit-
14 able to model MSPSP by MM-RCPSP, since the number of modes for tasks increases explosively.
15 Therefore the branch and bound methods proposed to solve RCPSP or MM-RCPSP (see Min-
16 gozzi et al. (1998) and Brucker et al. (1999) for more details) are not practical for MSPSP.
17 Below, we list the main solution approaches proposed in the literature:
18
19

20 - *Exact methods*: Branch and bound (Bellenguez and Neron 2007, Buddhakulsomsiri and
21 Kim 2006, Heilmann 2003),

22 - *Heuristic methods*: MIP-based combinatorial approach (Hurkens 2009), Tabu search
23 (Drezet and Billaut 2008), Genetic algorithm (Wu and Sun 2006, Yoshimura et al. 2006, Valls
24 et al. 2009, Cai and Li 2000), Constraint programming based hybrid approach (Li and
25 Womer 2009), MIP-based approach (Heimerl and Kolisch 2009), (so-called) High-performance
26 local search heuristic (Estellon et al. 2009).
27

28 - *Meta-heuristic methods*: Priority-based rules, ant colony optimization and genetic algo-
29 rithm (Gutjahr et al. 2008), adaptive large neighborhood search with several destroy and repair
30 methods (Cordeau et al. 2009).
31
32

33 3.5 Other solution approaches for our problem

34
35 Cordeau et al. (2009) participated in the 2007 ROADEF Challenge and tied for second place
36 with Estellon et al. (2009). Cordeau et al. (2009) present a mathematical model for the precise
37 definition of our problem and report that it can not be solved for large instances optimally
38 in reasonable time. They develop a meta-heuristic method that consists of a construction
39 heuristic and an adaptive large neighborhood search with several destroy and repair methods.
40 The solution strategy is viewed as a standard simulated annealing algorithm with a complex
41 neighborhood search due to the acceptance criterion of the solutions. Their results are included
42 in the computational results section.
43
44

45 Estellon et al. (2009) designed a local search scheme in which a greedy algorithm is employed
46 to obtain a feasible solution and this solution is improved by a local search strategy. The authors
47 use a methodology including three key points, *search strategy*, *moves* and *evaluation of moves*.
48 They also point out that a careful implementation increases the convergence speed of local-
49 search heuristics and stochastic elements are useful to improve the diversification.
50

51 As mentioned in the introduction, Hurkens (2009) considers the same problem and proposes
52 a two-phase MIP-based solution methodology. In the first phase, a MIP model determines
53 the tasks to be outsourced and calculates lower bound for the schedule cost discarding the
54 outsourced tasks. The lower bounds are found by relaxing skill requirements and precedence
55 relations of tasks and by allowing pre-emption. In the second phase, a strategy of assigning
56 simultaneously multiple tasks and technicians is followed. A partially constructed schedule is
57 extended by that assignment strategy. The assignment problem is formulated as an integer
58 linear programming model.
59
60
61
62

4 Constructing schedules with flexible matching models

A preparation phase is conducted to calculate the necessary parameters to be used in the construction phase. Our combinatorial algorithm builds day schedules. Construction of a day schedule starts with finding many-to-one type technician-task assignments. We call this initialization “*initial matching*”. The initialized partial schedule includes teams with single tasks in their workloads. Next, workloads are increased by adding more tasks and reallocating technicians to teams. A team can receive tasks until either the total duration of its workload reaches the length of a workday or the remaining tasks can not be performed due to skill requirements or precedence relations. Initial matching and extension of partial day schedule is performed by using two mixed integer linear programming models. The first one is called the “Initial Matching Model” or the IMM and the second one, called the “Flexible Matching Model” or the FMM, extends the schedule by iteratively adding more tasks and adapting the teams flexibly.

4.1 Preparation Phase

Before we construct schedules, we conduct some preprocessing steps and compute lower bounds for schedule cost and make the outsourcing decision.

4.1.1 Preprocessing of the data for a single task

Let MT_j denote the minimum number of technicians required by task j and let MH_j denote the man-hour demand of task j . It is necessary to know MT_j for two reasons. Firstly, lower bounds are calculated using man-hour demands of tasks and we have $MH_j = MT_j d_j$. Secondly, efficiency of assignments during schedule construction is maintained by punishing the excess in number of technicians from MT_j . A simple integer programming model is used to calculate MT_j . The binary decision variable x_t becomes 1 if technician t is assigned to the task j and 0 otherwise. The constraints of the model enforce that skill requirements of task j are met.

$$MT_j = \min \left\{ \sum_{t \in T} x_t : \sum_{t \in T} SM_t^{(l,s)} x_t \geq RQ_j^{(l,s)}, \forall (l, s) \in \mathbb{L} \times \mathbb{S}; x_t \in \{0, 1\} \right\}$$

A task is said to be *hard* if its required skills are not common among technicians. Hence hardness of a task depends on the skill distribution in the technician group as well as the required skill levels. For example a task requiring moderate skill levels may be “relatively” hard for a technician group, if there are few technicians specialized in the fields of demand. Let $SM_T = \sum_{t \in T} SM_t$ be the total skill of the technician group. Then the relative hardness h_j of task j is given by the following:

$$h_j = \max \left\{ \frac{RQ_j^{(l,s)}}{SM_T^{(l,s)}} \mid (l, s) \in \mathbb{L} \times \mathbb{S} : SM_T^{(l,s)} \neq 0 \right\}$$

Note that $0 \leq h_j \leq 1$ for feasible problem instances. The ratio $RQ_j^{(l,s)}/SM_T^{(l,s)}$ has small value for common skill levels among technicians. The value of h_j gives a sign for the relative expertise requirement and the extreme case $h_j = 1$ shows that task j can not be performed when certain experts are unavailable. We give the following definition for such tasks:

Definition A task j with $h_j = 1$ is called a “special” task.

Task j is said to *likely cover* task k if the expertise required by task k but not required by task j , is common within the technician group. Then the possibility that the team having task j in its load can also perform task k is high. Let $\alpha_{jk}^{(l,s)}$ be the pairwise comparison of skill requirements of tasks j and k for skill level (l, s) . Pairwise coverage relations are defined between the tasks if the sum of their durations is less than or equal to a workday length. Note that if task j dominates the skill requirement of task k or $\alpha_{jk}^{(l,s)} \geq 0$ for all (l, s) , then task j covers task k . If we have $RQ_j^{(l,s)} < RQ_k^{(l,s)}$ for some (l, s) and if these skills are common within the technician group, then it is more likely that a team performing task j can also perform task k . Pairwise comparison for each skill level between task j and k is given as follows:

$$\alpha_{jk}^{(l,s)} := \begin{cases} 0 & \text{if } RQ_j^{(l,s)} = RQ_k^{(l,s)} = 0 \\ 1 & \text{if } RQ_j^{(l,s)} \geq RQ_k^{(l,s)} \text{ and } RQ_j^{(l,s)} \neq 0 \\ \frac{SM_T^{(l,s)}}{(SM_T)_{\max}} & \text{if } RQ_j^{(l,s)} < RQ_k^{(l,s)} \end{cases}$$

where $(SM_T)_{\max} = \max_{(l,s) \in \mathbb{L} \times \mathbb{S}} \{SM_T^{(l,s)}\}$. The coverage of task j over task k is accepted, if the average value of entry-wise comparisons is greater than 0.9:

$$\gamma_{jk} := \begin{cases} 1 & \text{if } \frac{\sum_{(l,s)} \alpha_{jk}^{(l,s)}}{\sum_{(l,s)} \text{sign}(\alpha_{jk}^{(l,s)})} > 0.9 \\ 0 & \text{otherwise} \end{cases}$$

Finally the coverage of task j is given below:

$$cov_j = \sum_{k \neq j, d_j + d_k \leq H} \gamma_{jk}$$

The proposed combinatorial algorithm treats tasks in decreasing order of their weights. The weight of a task j is a combined measure of the following criteria: hardness h_j , coverage cov_j , resource use MT_j , duration d_j and precedence relations. The weight function in a general form we use is as follows:

$$w_j = \mathbf{A}(\overline{h_j} + \overline{cov_j} + \overline{MT_j} + \overline{d_j}) + \mathbf{B} \overline{MT_j} \overline{d_j} + \mathbf{C} \sum_{k:j \rightarrow k} \overline{MT_k} \overline{d_k} + \mathbf{D} \overline{h_j} \overline{cov_j}$$

where “ $\overline{}$ ” is used to interpret that all criteria are normalized. Values of coefficients \mathbf{A} , \mathbf{B} , \mathbf{C} and \mathbf{D} are determined in such a way that contributions of all expressions have the same order of magnitude. In the combinatorial algorithm, a task of high weight is selected with high probability.

4.1.2 Lower bounds

Lower Bounds for RCPSP are found in the literature by several methodologies. Some of them are linear programming based lower bounds (Mingozzi et al. 1998), destructive approach (Heilmann and Schwindt 1997), Lagrangian relaxation of the corresponding integer programming formulation (Möhring et al. 2003) and destructive approach combined with constraint programming techniques (Brucker and Knust 2000). The mentioned lower bound techniques for RCPSP are not directly applicable to our problem because of its special aspects. We find lower bounds of our problem by solving a simplified problem which is formulated as a MIP model. This method was first proposed by Hurkens (2009) and in this section we provide a brief explanation. For further details we refer to the author.

In the simplified problem, skill requirements and precedence relations are relaxed and the schedule is pre-emptive. The simplified problem amounts to finding the minimum time needed to meet the cumulative *man-hour* demand considering technician availabilities only. If we let $J(p) = \{j \in J | P(j) = p\}$ denote the subset of tasks in priority class p , then man-hour demand of tasks in $J(p)$ can be expressed as:

$$MH_{J(p)} = \sum_{j \in J(p)} MH_j$$

A MIP model, called the Lower Bound Model, is used to compute the minimum aggregated times by selecting the best choice of tasks to outsource. Note that outsourcing decreases man-hour demand of tasks.

In the real case we handle priority classes sequentially. Although the priority weights suggest the ordering $\{1, 2, 3, 4\}$, we consider all priority permutations of practice: $\{1, 2, 3, 4\}$, $\{1, 3, 2, 4\}$, $\{2, 1, 3, 4\}$, $\{2, 3, 1, 4\}$, $\{3, 1, 2, 4\}$, $\{3, 2, 1, 4\}$. It is important to note that depending on the number of tasks of priority classes any priority sequence may turn out to include the optimal schedule. The reason why priority class 4 is kept as the last one is that its weight is zero, so it has no effect on the objective of our problem.

4.2 Construction Phase

Having found lower bounds and the tasks to outsource in the preparation phase, the combinatorial algorithm proceeds to constructing alternative schedules for every priority permutation. The priority permutation with smallest lower bound value is considered first. Alternative schedules are constructed by changing several strategies and by varying several parameters in our matching models. The term “*matching*” denotes many-to-one type technician-task assignment. Two matching models used by combinatorial algorithm are formulations of many-to-one type technician-task assignments on a bipartite graph. The first (also simpler) one is used for initialization of day schedules. The second one extends an initialized day schedule by adding as many tasks as possible.

4.2.1 Initial Matching

Constructing a day schedule starts with parallel assignments of technicians to candidate tasks. Candidate tasks correspond to so-called eligible activities by Brucker et al. (1999). The parallel assignment of technicians to tasks is superior to selecting a single task and building a team for

Table 1: Sets, indices, parameters, and variables of the IMM

<i>Sets included in the model</i>	
J'	set of tasks, $J' \subseteq J$
T'	set of technicians, $T' \subseteq T$
<i>Indices</i>	
j	task index, $j \in J'$
t	technician index, $t \in T'$
<i>Parameters</i>	
w_j	weight of task j
$SM_t^{(l,s)}$	competence of technician t in domain s at level l
$RQ_j^{(l,s)}$	number of skilled technicians required by task j in domain s at level l
<i>Variables</i>	
x_{tj}	binary variable indicating whether technician t is assigned to task j
y_j	binary variable indicating whether task j initializes a team

that task, since parallel assignment has a more global view compared to the latter one. For example, the constructive heuristic of Cordeau et al. (2009) constructs a team for a selected seed task and continues building teams for other ones. In our opinion this assignment procedure misses the global view of skills compared to parallel assignment. The effect of starting with a global view of skills is more remarkable in problem instances where expertise is rare.

In the first part of the construction phase, the partial day schedule is initialized by finding parallel technician-task assignments. We create a bipartite graph where the left partition represents a subset of technicians and the right partition represents a subset of tasks (See Figure 1). Then the parallel technician-task assignments correspond to many-to-one type matchings. The problem of finding the simultaneous assignments is formulated as an integer linear programming model. The goal is to maximize the weighted number of team initializations. Weights of tasks represent their power to initialize a team.

In principle we aim to find the optimal initial matching on the complete set of available technicians and candidate tasks. Due to the time limit imposed by the problem definition we settle for a heuristic solution by repeatedly applying the IMM to a subset of technicians and a subset of tasks. The number of technicians and tasks is determined in such a way that the solution of the model can be obtained in a reasonably short time using a commercial solver like CPLEX 11.0. Therefore the initial matching model is run several times if necessary. Note that after the initial matching the partial day schedule consists of teams with a single task in their load.

Table 1 shows the notation, parameters and decision variables of the IMM. The mathematical formulation of the IMM is as follows:

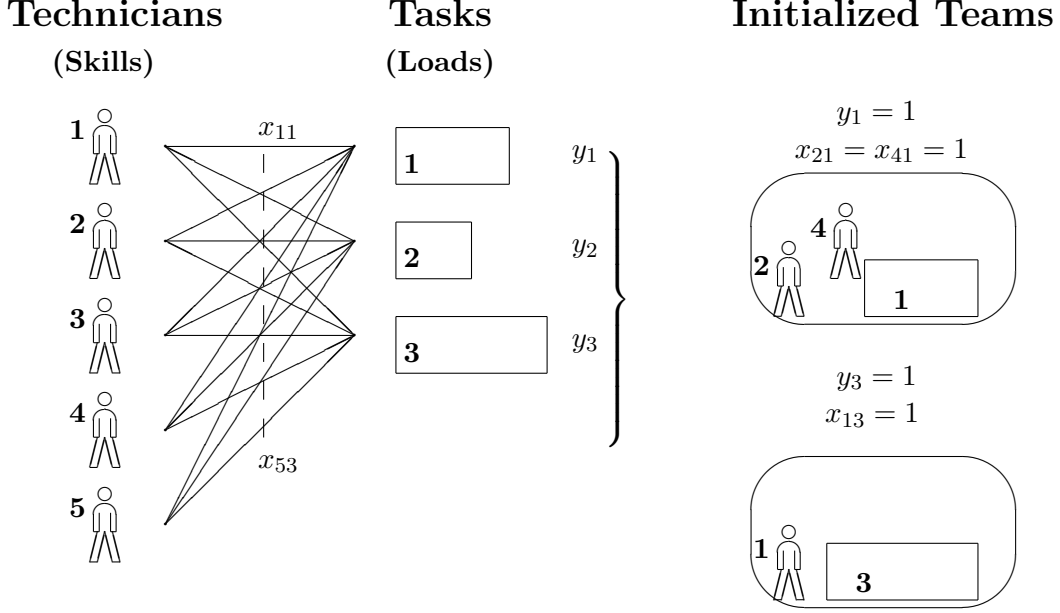


Figure 1: Initializations of teams in the IMM

$$\text{maximize } \sum_{j \in J'} w_j y_j$$

subject to:

$$\sum_{j \in J'} x_{tj} \leq 1 \quad \forall t \in T' \quad (1)$$

$$\sum_{t \in T'} SM_t^{(l,s)} x_{tj} \geq RQ_j^{(l,s)} y_j \quad \forall (l,s) \in \mathbb{L} \times \mathbb{S}, \forall j \in J' \quad (2)$$

$$y_j, x_{tj} \in \{0, 1\} \quad \forall j \in J', \forall t \in T'$$

Constraints (1) enforce that a technician can be matched to at most one task. If a task initializes a team, then constraints (2) guarantee that it must be matched to a number of technicians such that its skill requirements are met. Note that a task with a relatively high weight may get more technicians than necessary. Although it might seem as an inefficiency, in the extension phase the FMM reallocates technicians in order to achieve the possible maximum weighted packing of tasks in the day schedule. When no technician is left for assigning to tasks or all candidate tasks are assigned to technicians, the initial matching is completed. The initialized day schedule has teams with single tasks in their workloads.

Figure 1 illustrates team initializations with an example IMM of 5 technicians and 3 tasks. According to the solution of the model, it turns out that 2 teams are initialized by j_1 and j_3 . In the partial schedule team τ_1 (τ_2) has load with duration d_1 (d_3). After the initial matching,

1
2
3
4
5
6
7
8 the partial schedule has the following technicians groups $T(\tau_1) = \{t_4, t_2\}$ and $T(\tau_2) = \{t_1\}$.
9 Schedule construction will continue with extending the partial schedule by:

- 10 - adding more tasks to the initialized teams
- 11 - merging similar teamloads to let some technicians contribute to other teams
- 12 - letting some tasks initialize teams
- 13 - recombining the technicians of teams to adapt to the new skill requirements

14
15 The following subsections explain what kind of MIP model we use for extending the partial
16 schedule and how the mentioned cases may occur.
17
18

19 **4.2.2 Extending the partial day schedule**

20
21 After a day schedule has been initialized, it is extended by iteratively scheduling more tasks.
22 Adding more tasks to the schedule is not trivial, since some tasks may require slightly or even
23 completely different skills than the skills of available teams. Therefore more tasks can be added
24 to the partial schedule if technicians are easily reallocated among teams depending on the skill
25 requirements of candidate tasks. Rearranging technician combinations of teams is achieved by
26 letting their technicians be *conditionally* available.
27
28

29 *Generalization from “technicians and tasks” to “skills and loads”:*

30 In a partial day schedule a team can be considered from different points of view. First, a
31 team can be perceived as a combination of *skills* if technicians are considered. Second, a team
32 can also be perceived as a *load* if its workload is considered. Therefore from the first (second)
33 point of view a team represents a *skill* (*load*). Moreover unassigned individual technicians
34 (candidate tasks) can be perceived as *skills* (*loads*). These observations lead us to consider
35 every element of the partial schedule as “skill” and as “load”. Skills and loads are grouped as
36 shown in Table 2. Instead of technicians and tasks we will have skills and loads in the matching
37 model to extend the partial day schedule.
38
39

40 The skills and the loads of teams are called “active”, since they are actively comprising the
41 partial schedule. The technicians who were not assigned in the partial schedule to any task
42 are called “passive” skills. Similarly, the candidate tasks are called “passive” loads. Individual
43 technicians of team are called “latent” skills. As mentioned above, latent skills are conditionally
44 available and they play an important role in scheduling candidate tasks (passive skills) by
45 determining skill combinations of loads flexibly.
46

47 Similar to the IMM, we define a bipartite graph in which the left partition includes skills
48 and the right partition includes loads. In this bipartite graph, skill-load assignments correspond
49 to many-to-one type matchings. Teams are represented in the left partition by their skills and
50 in the right partition by their loads. In addition we have skills of unassigned technicians in the
51 left partition and loads of candidate tasks in the right partition. The possible extension cases
52 that can appear in a solution are listed below:
53
54
55
56
57
58
59
60
61
62
63
64
65

Table 2: Types of Skills and Loads

<i>Skills</i>	
<i>active</i>	total technician skills in a team
<i>passive</i>	unassigned technician
<i>latent</i>	conditionally available technician
<i>Loads</i>	
<i>active</i>	workload of a team
<i>passive</i>	candidate task

- Case 1** an active skill is matched with:
- a) an active load: *merging*
 - b) a passive load: *extending* team load
- Case 2** an active load is matched with:
- a) passive and/or latent skills: *recombining* technicians
 - b) skills including an active skill: *merging*
- Case 3** a passive load is matched with:
- a) passive and/or latent skills: *initializing* a team
 - b) skills including an active skill: *extending* team load

As seen above a load can be matched with *any* skill combination as long as the current schedule configuration allows. This is the key aspect of our matching model resulting in high flexibility. Each technician, no matter in a team or not, becomes a potential skill for candidate tasks. Candidate tasks may be added to the partial schedule by joining a teamload (Case 1a) or by initializing a team (Case 3a). While joining to a teamload, a candidate task may bring some additional technicians to the team, if necessary.

The condition for a latent skill to be in other matchings is that the active load of its team is matched with skills. In other words, matching an active load with skills leads to an opportunity of using some of its technicians in other matchings. Generally speaking, matching of active loads to skill combinations is triggered by passive loads (candidate tasks), since they mainly contribute to the objective of matching model. The higher weight a candidate task has, the more power to force the current partial schedule to find the needed skill combination. In light of this fact we have the following observation:

Observation *A candidate task with sufficiently high weight may force teams to merge or recombine their technicians, thereby making an expert technician available.*

Case 1a, merging, generally leads to an increase in overall efficiency of the partial schedule. For example, let two tasks with similar skill requirements have initialized teams in the initial matching. After merging in the extension phase they can be performed by one technician group and the second technician group can be used in initializing teams or extending loads of other teams. Figure 2 and Figure 3 illustrate examples of extending the partial schedule of the example in Figure 1. In order to show all possible cases, we illustrated two different scenarios corresponding to two different solutions obtained with different parameter settings.

In Figure 2 (and 3) the flexible matching model includes 2 teams (τ_1, τ_2) with technicians $T(\tau_1) = \{t_4, t_2\}$ and $T(\tau_2) = \{t_1\}$, 2 unassigned technicians (t_3, t_5) and one candidate task (j_2). Latent skills are shown in black boxes and linked to their current teams. The first two indices

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

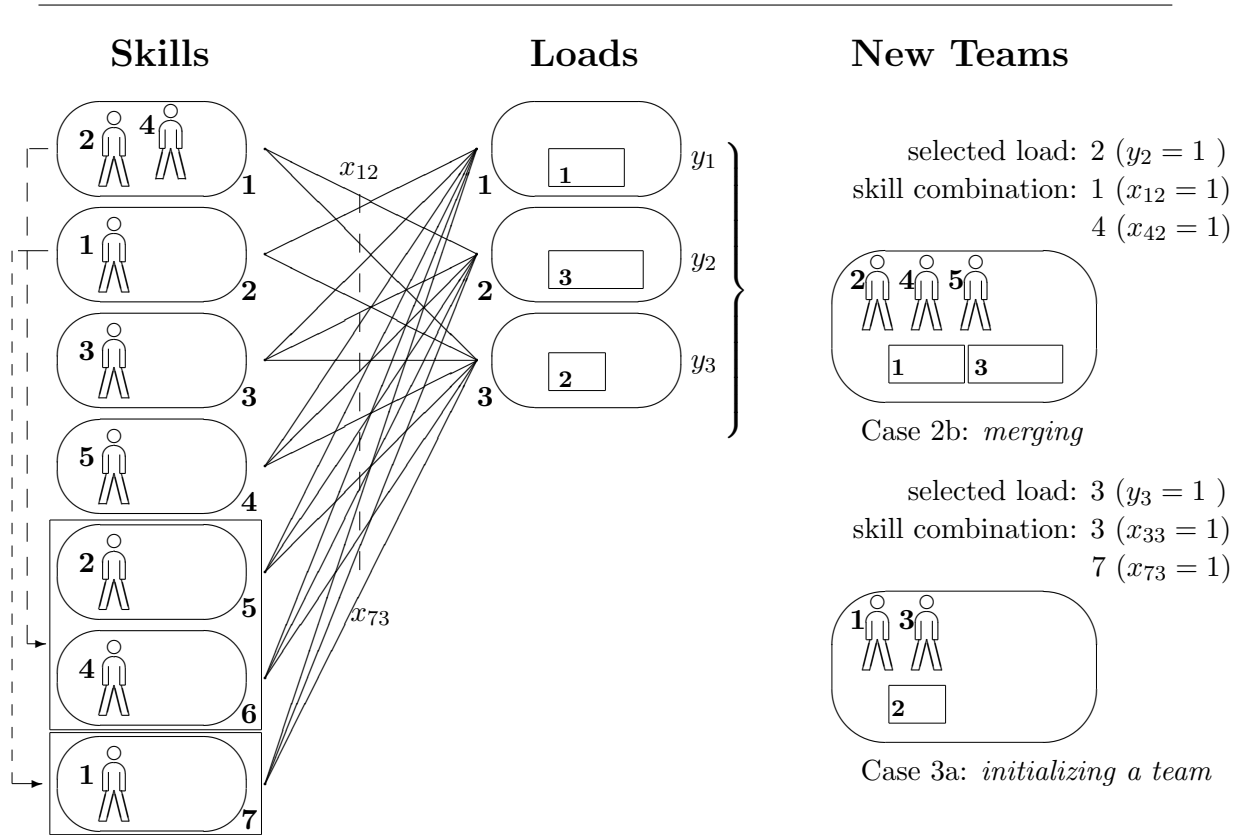


Figure 2: Extending partial schedule with the FMM (Scenario 1)

of partitions belong to active skills and loads. There is no edge drawn between an active skill and an active load of the same team, since this matching does not make any sense and it is forbidden in the model.

In scenario 1 (Figure 2), the active load of τ_2 is matched to a new skill combination including the active skill of τ_1 and passive skill of t_5 . This matching is an example of merging (Case 1a) and the combined load is performed by the newly matched skill combination. Matching active load of τ_2 allowed t_1 to be in the initialized team of j_2 (Case 3a) together with technician t_3 .

In scenario 2 (Figure 3), τ_1 recombined its technicians (Case 2a) by being matched to passive skills of unassigned technicians t_3 , t_5 and latent skill of technician t_2 who was in the previous combination as well. The number of technicians of τ_1 seems increased by one and this increase can be explained as an adjustment to have technician t_4 in the extended team τ_2 (Case 1b) as we now have j_2 in one load.

Due to its mentioned flexibility aspects, we call the bipartite matching model the *flexible matching model*, or the FMM. The main contribution of this study is introducing this model. Hurkens (2009) extends the partial day schedule using two different matching models iteratively, where the first one is used to assign multiple tasks simultaneously, and the second one is used to find efficient recombination of technicians. The FMM contains both steps and more.

Forbidden edges:

In the FMM some edges may cause infeasibility and some are useless. Those edges are detected in advance and they are not allowed to be in any matching. These edges can be listed

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

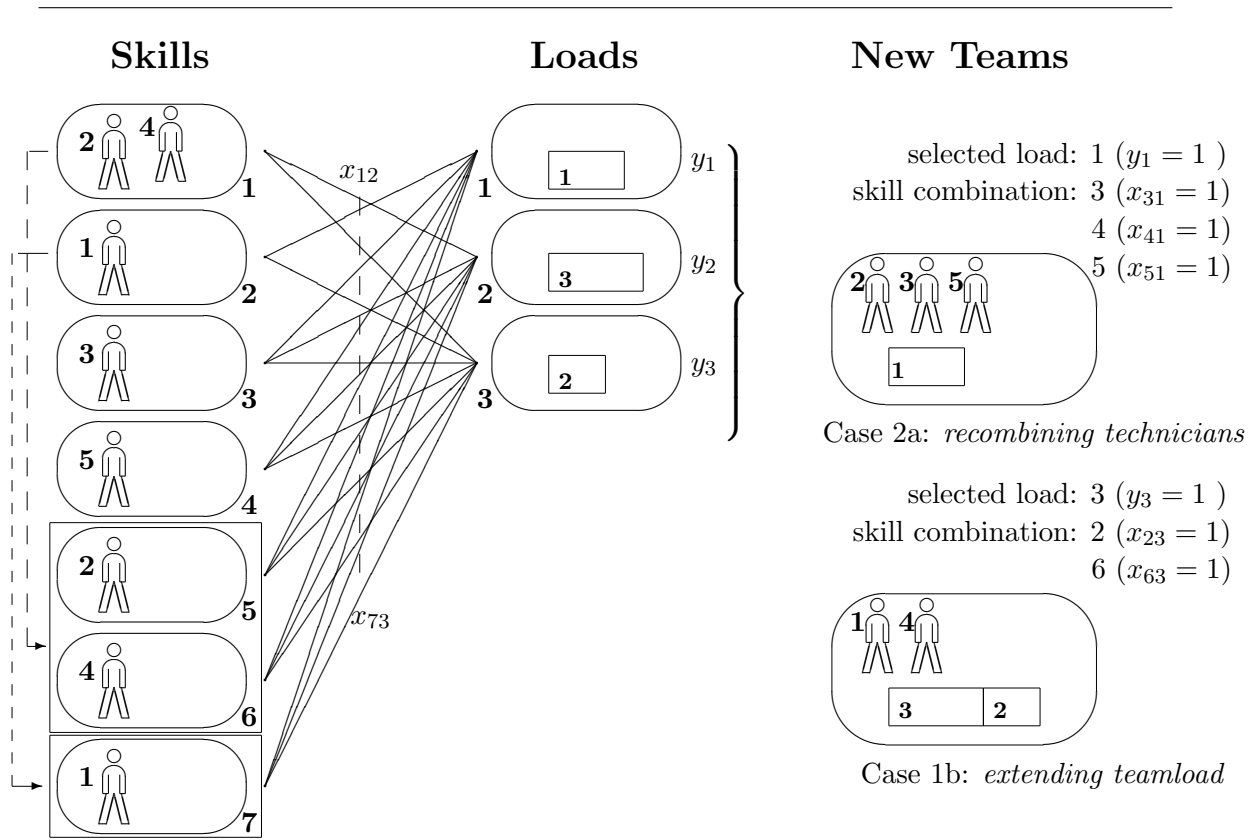


Figure 3: Extending partial schedule with the FMM (Scenario 2)

Table 3: Sets, indices, parameters and variables of the FMM

<i>Sets included in the model</i>	
Δ	set of all skills in the left partition
Λ	set of all loads in the right partition
\mathbb{T}	set of active skills
\mathbb{W}	set of team loads
\mathbb{F}	forbidden edges, $\mathbb{F} \subseteq \Delta \times \Lambda$
\mathbb{M}	set of edges between skill and load of the same team
<i>Indices</i>	
σ, σ'	skill index, $\sigma, \sigma' \in \Delta$
λ, λ'	load index, $\lambda, \lambda' \in \Lambda$
<i>Parameters</i>	
$T(\sigma)$	set of latent skills of active skill σ
w_λ	weight associated to load λ
mt_λ	the number of technicians used for performing all tasks in load λ
$SM_\sigma^{(l,s)}$	skill value of σ in domain s at level l
$RQ_\lambda^{(l,s)}$	number of skilled technicians required by λ in domain s at level l
<i>Variables</i>	
$x_{\sigma\lambda}$	binary variable indicating whether σ is assigned to λ
y_λ	binary variable indicating whether λ assigned to a skill combination
z_λ	inefficiency penalty of λ

as follows:

- the edges between active load and active skill of the same team,
- the edges that lead to a total duration of a teamload longer than a work day
- the edges that result in violation of a precedence relation
- the edges that cause a delay in an already completed priority class

Punished matchings:

The number of teams included in the FMM is limited. Therefore a matching although it seems profitable for the subset of the partial schedule may not be so for the complete partial schedule. The immediate example is a matching of active skill and passive load in which an active skill (technician group of a team) has more technicians than needed for the passive load (candidate task). The FMM has penalty variables to detect and prevent these matchings.

4.2.3 Mathematical formulation of the FMM (Flexible Matching Model)

The FMM finds many-to-one type assignments on a bipartite matching model whose left (right) partition represents *skills* (*loads*). The sets, indices, parameters and variables are listed in Table 3.

Note that $|\mathbb{T}| = |\mathbb{W}| = |\mathbb{M}|$, since each team has its skill in \mathbb{T} and its load in \mathbb{W} with corresponding indices $(\sigma, \lambda) \in \mathbb{M}$. The weight function of loads is defined as follows:

$$w_\lambda := \begin{cases} 0 & \text{if } \lambda \in \mathbb{W} \\ w_j & \text{if } \lambda \notin \mathbb{W} \text{ where } j \in J \text{ is the single task of load } \lambda \end{cases}$$

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

$$\text{maximize} \quad \sum_{\lambda} w_{\lambda} y_{\lambda} - z_{\lambda}$$

subject to:

$$\sum_{\lambda} x_{\sigma\lambda} \leq 1, \quad \forall \sigma \in \Delta \quad (3)$$

$$\sum_{\sigma \in \mathbb{T}} x_{\sigma\lambda} \leq 1, \quad \forall \lambda \in \Lambda \quad (4)$$

$$x_{\sigma\lambda} = 0, \quad \forall (\sigma, \lambda) \in \mathbb{F} \quad (5)$$

$$\sum_{\lambda'} x_{\sigma\lambda'} + y_{\lambda} \leq 1, \quad \forall (\sigma, \lambda) \in \mathbb{M} \quad (6)$$

$$\sum_{\sigma' \in T(\sigma)} \sum_{\lambda'} x_{\sigma'\lambda'} \leq |T(\sigma)| y_{\lambda}, \quad \forall (\sigma, \lambda) \in \mathbb{M} \quad (7)$$

$$\sum_{\sigma} SM_{\sigma}^{(l,s)} x_{\sigma\lambda} \geq RQ_{\lambda}^{(l,s)} y_{\lambda}, \quad \forall (l, s) \in \mathbb{L} \times \mathbb{S}, \forall \lambda \in \Lambda \quad (8)$$

$$\sum_{\sigma \in \mathbb{T}} |T(\sigma)| x_{\sigma\lambda} + \sum_{\sigma \in \Delta \setminus \mathbb{T}} x_{\sigma\lambda} - mt_{\lambda} \leq z_{\lambda}, \quad \forall \lambda \in \Lambda \quad (9)$$

$$x_{\sigma\lambda}, y_{\lambda} \in \{0, 1\}, z_{\lambda} \geq 0$$

A skill can be assigned to at most one load (constraints (3)). A load may be matched to at most one active skill according to constraints (4). It can be matched to any number and any combination of passive and latent skills though. Having constructed the set \mathbb{F} in advance, edges in \mathbb{F} are forbidden by constraints (5).

In the FMM, a team has an active skill $\sigma \in \mathbb{T}$ and latent skills $T(\sigma)$ in the left partition and has a load $\lambda \in \mathbb{W}$ in the right partition. According to constraints (6), a team can contribute to extending the day schedule in one of the following ways: either its active skill is matched to a load or its active load is matched to skills. In the former case, the matched load is added to team's workload and technicians of the team stay together. Some additional technicians may join to team as well, if some other latent or passive skills are also assigned to matched load. In the latter case, a skill combination is assigned to active load (constraints (8)) and latent skills of the team may be used in other matchings (constraints (7)). Inefficiency variables z_{λ} try to avoid the assignments that have an unnecessarily high number of technicians (constraints (9)). Here $mt_{\lambda} = MT_j$ for a passive load λ with candidate task j and $mt_{\lambda} = |T(\sigma)|$ for an active load λ currently assigned to σ . For instance, if a load of a candidate task with $MT = 2$ is assigned to a team skill of 5 technicians, then this assignment is penalized by constraints (9) on the value of z_{λ} . Note that inefficiency is not forbidden in the FMM, but discouraged by penalizing.

The objective function is the sum of the weights of selected loads and the inefficiency drop. The candidate tasks contribute the objective by their weights and influence the allocation of skills among teams. Note that a task with sufficiently high weight can even cause some inefficiency to get scheduled.

4.2.4 Strategies applied in constructing alternative schedules

In this section we explain the strategies applied to find alternative schedules. We have observed in experiments that each of those strategies may lead to a best solution.

Efficiency in Initial Matching: When the formulation of the FMM is carefully examined, it is not difficult to see that if $\mathbb{T} = \emptyset$, the FMM boils down to the IMM plus efficiency constraints. So one of our strategies is adding efficiency constraints to the Initial Matching. This strategy is especially beneficial in instances where the average duration of tasks is close to workday length.

Fully loaded teams: While extending the partial day schedule some teams may reach a workload with total duration of a workday. As a strategy, we include a few of those fully loaded teams in the FMM in the hope to lower their skills excess by making some of their technicians conditionally available for candidate tasks. Fully loaded teams can contribute to extending partial schedule by recombining their technicians.

Number of teams in the FMM: In order to obtain the optimal solution of the FMM in a reasonably short time, a fixed number of teams are included. So we have $|\mathbb{T}| = |\mathbb{W}| \in \{3, 4, 5, 6, 7, 8\}$. As long as time limit allows, we construct schedules for each fixed number. In large instances usually a few of them can be used.

Selection of candidate tasks: Candidate tasks are included in the FMM in sequential order according to their priority classes. As a strategy we allow some of the special tasks in succeeding priority classes to be in the FMM. This way we aim to avoid the delays of the priority makespan due to rare expertise.

4.3 An overview of the combinatorial algorithm

In the preparation phase, key properties of tasks are measured and a lower bound value of each priority permutation is calculated. The construction phase starts building alternative schedules from the priority permutation having the smallest lower bound. Throughout the construction phase we keep track of the schedule having lowest cost among the ones previously constructed. This incumbent schedule is updated whenever a schedule with lower cost is encountered. If the cost of the incumbent schedule is lower than or equal to the lower bound value of a priority permutation to which the algorithm just proceeded, then that priority permutation is neglected. It is clear that no schedule constructed under that priority permutation can have lower cost than the cost of the incumbent schedule. Since the priority permutations are considered in nondecreasing order with respect to their lower bound values, neglecting a priority permutation means that the remaining priority permutations should also be neglected. So the combinatorial algorithm finalizes schedule construction and outputs the incumbent schedule.

Alternative schedules are constructed by varying some parameters and changing strategies in our matching models within the allowed time. The flowchart of the combinatorial algorithm can be seen in Figure 4. After the preparation phase, constructing alternative schedules starts by picking a priority permutation with smallest lower bound value. Next the maximum number of teams in the FMM and the strategies to follow are chosen. A partial schedule is started by running the IMM and it is extended iteratively using the FMM. If no solution of the FMM contains a matching, then we finalize the day and check whether all tasks are scheduled. If there

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

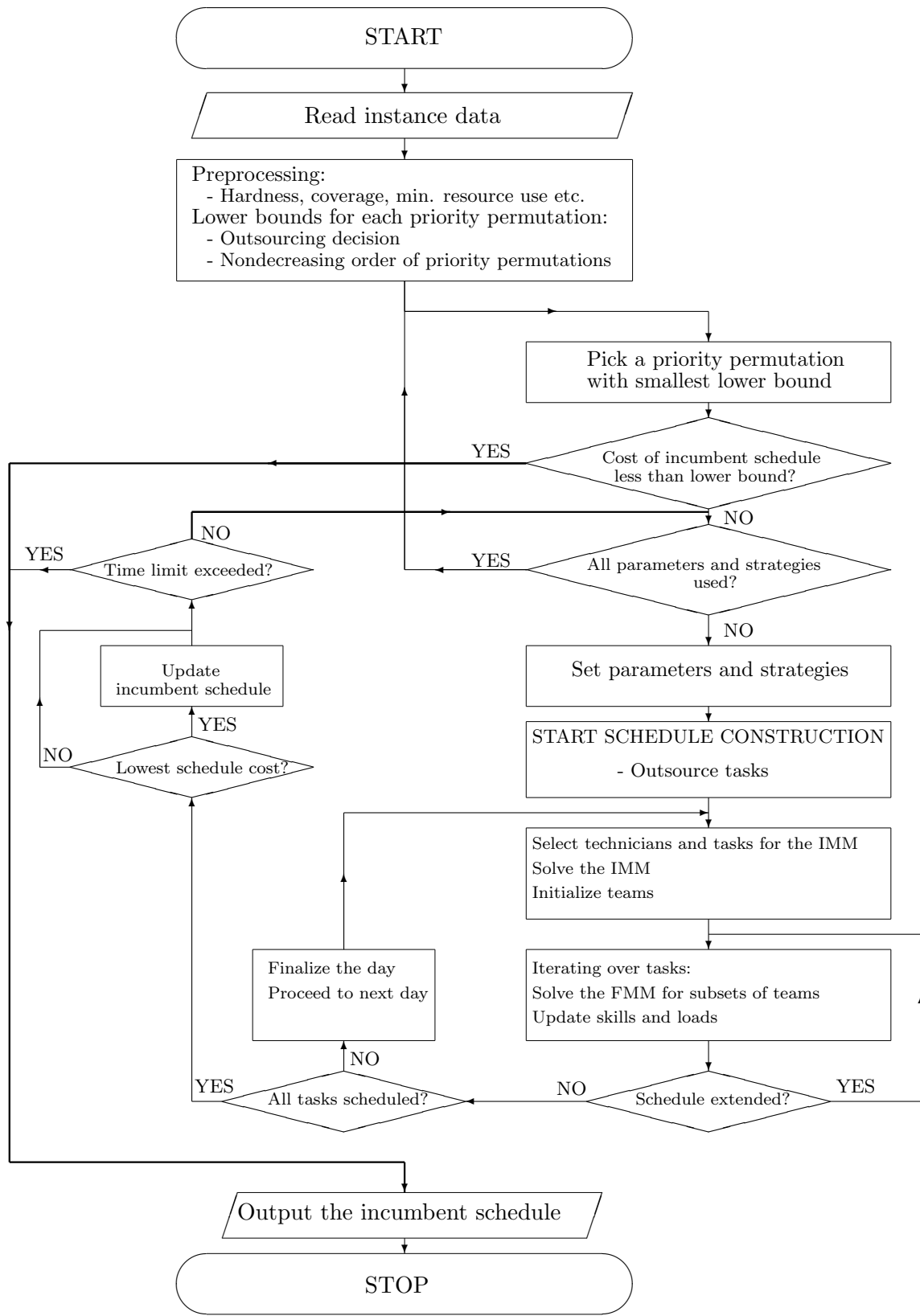


Figure 4: Flowchart of the Combinatorial Algorithm

1
2
3
4
5
6
7
8 are more tasks to schedule, then the next day starts. Otherwise the schedule is finalized and
9 compared to the incumbent schedule. The incumbent schedule is updated if the newly found
10 schedule has lower cost. The construction phase ends when either the time limit is exceeded
11 or all promising priority permutations have been considered. Finally the incumbent schedule is
12 given as output.

13 Cordeau et al. (2009) and Estellon et al. (2009) argue that the combinatorial algorithm pro-
14 posed by Hurkens (2009) is an application of local search with large neighborhood exploration.
15 However both in our algorithm and Hurkens’ algorithm, alternative schedules are constructed
16 making use of MIP models and the constructed schedules are not modified by any destruct and
17 repair methods with the hope to improve the solution quality. The strategy of both algorithms is
18 to obtain good quality solutions with the simultaneous technician-task assignments. Therefore
19 both algorithms can not be classified as local search algorithms, rather they are constructive
20 heuristics employing MIP models for simultaneous technician-task assignments.
21
22
23

24 5 Computational results

25 5.1 On the rare expertise in problem instances

26
27 Three sets of problem instances were provided by France Telecom in the 2007 ROADEF Chal-
28 lenge. The descriptive statistics of the instances can be seen in Table 4. The number of skill
29 domains (levels) varies from 3 to 40 (2 to 7). Data set A includes relatively small instances
30 compared to B and X. There is no remarkable difference between data set B and X in terms of
31 number of technicians and number of tasks, however those data sets do differ in the number of
32 special tasks (for definition see section 4.1.1).
33
34
35

36 The costs of schedules constructed by the FMM and other heuristics are given in Table 5.
37 The first column shows the problem instance and the next four columns report the results found
38 by FMM, Hurkens (2009), Cordeau et al. (2009) and Estellon et al. (2009). In each column,
39 the first entries are schedule costs and second entries are the relative difference in percentage
40 that is defined as the difference of a schedule cost to the best schedule cost ever found (note
41 that best schedules are listed in the column with title “N. BEST”). The column with label “O.
42 BEST” shows the best results found in the evaluation of the 2007 ROADEF Challenge and “N.
43 BEST” reports the ever best schedules that we are aware of. The last column labeled “LB” list
44 the lower bound values for each instance.
45
46

47 It is seen in the last column of Table 4 that data set X includes instances with a higher
48 number of special tasks. The average number of special tasks of the instance groups A, B and X
49 are 1.3, 2.4 and 7.2 respectively. This hints that data set X instances have a heterogeneous skill
50 distribution among technicians and therefore rare expertise is observed. The challenge in the
51 instances of rare expertise can be realized by checking the gap between the best found schedules
52 and lower bounds. In Table 5 we see that the gap between best schedules and lower bounds is
53 smaller in data sets A and B compared to data set X. This may show either the weakness of
54 lower bounds or the situation that the reported approaches are not much successful in handling
55 rare expertise or maybe both. However it is remarkable that most of the best schedules (7 out
56 of 10 in the O. BEST column) of X instances are due to Hurkens (2009). It can be concluded
57 that the combinatorial algorithm introduced by Hurkens (2009) is promising for rare expertise
58 instances. If the N.BEST column is examined, it is seen that the FMM found 7 best schedules
59
60
61
62

Table 4: Problem instances A, B and X

Instance	Data set A					Data set B					Data set X				
	$ T $	$ J $	$ S $	$ L $	$ SP $	$ T $	$ J $	$ S $	$ L $	$ SP $	$ T $	$ J $	$ S $	$ L $	$ SP $
1	5	5	3	2	0	20	200	4	4	0	60	600	15	4	27
2	5	5	3	2	0	30	300	5	3	0	100	800	6	6	0
3	7	20	3	2	1	40	400	4	4	0	50	300	20	3	0
4	7	20	4	3	0	30	400	40	3	15	70	800	15	7	0
5	10	50	3	2	1	50	500	7	4	9	60	600	15	4	13
6	10	50	5	4	5	30	500	8	3	0	20	200	6	6	3
7	20	100	5	4	1	40	400	4	4	0	50	300	20	3	0
8	20	100	5	4	0	150	800	10	4	0	30	100	15	7	5
9	20	100	5	4	3	60	120	5	5	0	50	500	15	4	10
10	15	100	5	4	2	40	120	5	5	0	40	500	15	4	14

SP: Special Tasks, definition of a special task can be found in Section 4.1.1.

in data set X and 6 of them are among Hurkens’ best schedules. In our opinion this is the result of the increased flexibility in our combinatorial algorithm. The flexibility in finding new skill combinations leads to more efficient packing of special tasks and higher utilization of the rare expertise. Moreover in some instances like X9 and X10 the gap between best schedule and lower bound is decreased remarkably. As an improved version, the FMM found better schedules in 9 X instances compared to Hurkens’ combinatorial algorithm.

If all data sets are considered, it seems reasonable to conclude that data sets B and X are better representatives of the real case instances due to the high number of technicians, tasks and skill domains. Moreover rare expertise is also a situation companies encounter in their operations. In instances with a small number of technicians, tasks and skill domains, schedule costs are sensitive to individual assignments, whereas in large instances the number of feasible schedules are high and the schedules are not sensitive to individual assignments. Therefore in our opinion, large instances are better to test the reliability of the algorithms.

5.2 On the performances of heuristics

It is remarkable that the FMM has an average difference of 1.3% in data set X whereas it is 5.9%, 12.3% and 14.5% for Hurkens (2009), Cordeau et al. (2009) and Estellon et al. (2009) respectively. Cordeau et al. (2009) outperforms the other algorithms in data set A, however their average distance shows a high increase rate from A to B and from B to X. Hurkens (2009) has a slight increase from A to B and stays almost at the same level from B to X. The FMM starts with the “highest” average distance in data set A and draws a slight increase from A to B. However in both B and X it outperforms the others. The decrease in gap from B to X shows that in the instances of rare expertise, the FMM can find compact schedules and experiences less skill excess in assignments. Particularly, the instances X1, X5, X9 and X10 are sensitive to expert availabilities due to the high number of special tasks. Estellon et al. (2009) underline also those instances and emphasize the gap between the combinatorial approach by Hurkens (2009) and the other solution approaches.

Table 5: Results of problem instances A, B and X

Instance	FMM (%)		Hurkens (%)		Cordeau (%)		EsGaN0 (%)		O. BEST	N. BEST	LB
A1	2340	0.0	2340	0.0	2340	0.0	2340	0.0	2340	2340	2310
A2	4755	0.0	4755	0.0	4755	0.0	4755	0.0	4755	4755	2100
A3	12600	6.1	11880	0.0	11880	0.0	11880	0.0	11880	11880	11340
A4	14040	4.4	13620	1.2	13452	0.0	14040	4.4	13452	13452	10680
A5	30555	1.1	29355	0.0	29355	0.0	29400	0.2	29355	29355	26940
A6	20055	6.7	20280	7.9	18795	0.0	18795	0.0	18795	18795	17640
A7	31920	4.5	32520	6.5	30540	0.0	30540	0.0	30540	30540	28672
A8	17790	0.5	18960	7.1	17700	0.0	20100	13.6	17700	17700	16216
A9	29120	6.1	28320	3.2	27692	0.9	27440	0.0	27692	27440	25558
A10	40430	5.1	40650	5.7	38636	0.5	38460	0.0	38636	38460	36992
<i>Average</i>		<i>3.7</i>		<i>3.2</i>		<i>0.1</i>		<i>1.8</i>			
B1	34620	2.1	35460	4.6	37200	9.7	33900	0.0	34395	33900	31875
B2	17070	7.6	18300	15.3	17070	7.6	16260	2.5	15870	15870	14280
B3	16320	2.0	16965	6.0	18015	12.6	16005	0.0	16020	16005	13965
B4	26910	13.2	27015	13.6	23775	0.0	24330	2.3	25305	23775	16800
B5	92400	4.2	94200	6.2	117540	32.5	88680	0.0	89700	88680	79530
B6	29115	6.3	30510	11.4	27390	0.0	27675	1.0	27615	27390	24180
B7	32340	0.0	33060	2.2	33900	4.8	36900	14.1	33300	32340	25290
B8	35400	10.1	32160	0.0	33240	3.4	36840	14.6	33030	32160	31890
B9	28080	0.0	28080	0.0	29760	6.0	32700	16.5	28200	28080	25680
B10	35040	1.0	35040	1.0	35640	2.8	41280	19.0	34680	34680	32370
<i>Average</i>		<i>4.6</i>		<i>6.0</i>		<i>7.9</i>		<i>7.0</i>			
X1	147600	0.0	151980	3.0	159300	7.9	180240	22.1	151140	147600	136680
X2	7800	7.4	9090	25.2	8280	14.0	8370	15.3	7260	7260	5700
X3	48720	0.0	50400	3.4	50400	3.4	50760	4.2	50040	48720	36060
X4	64600	0.0	65640	1.6	66780	3.4	68960	6.7	65400	64600	58230
X5	145860	0.0	147000	0.8	157800	8.2	178560	22.4	147000	145860	130995
X6	9690	2.2	10440	7.7	9900	4.4	10440	10.1	9480	9480	6150
X7	34080	2.9	33120	0.0	47760	44.2	38400	15.9	33240	33120	25410
X8	23220	0.0	23580	1.6	24060	3.6	23800	2.5	23640	23220	17600
X9	128340	0.0	136020	6.0	152400	18.7	154920	20.7	134760	128340	98805
X10	122220	0.0	131700	7.8	140520	15.0	152280	24.6	137040	122220	87210
<i>Average</i>		<i>1.3</i>		<i>5.9</i>		<i>12.3</i>		<i>14.5</i>			
Overall		3.2		5.1		6.8		7.8			

6 Conclusions and future research

In this paper, we proposed a combinatorial algorithm that is using a Flexible Matching Model as a core engine for a multi-skill workforce scheduling problem. The scheduling problem was defined by France Telecom in the 2007 ROADEF Challenge. The opportunity of outsourcing some tasks is one aspect of our problem that distinguishes it from the similar ones defined in the literature. Technicians must work in a team for a workday and they use skills in all domains simultaneously while performing tasks. These properties of our problem make it not possible to use previously proposed methodologies without any modifications.

The main contribution of this study is introducing flexibility in the combinatorial algorithm that was firstly proposed by Hurkens (2009). Moreover we propose relative key measurements for tasks in a multi-skill requirement framework. The flexibility in matching models resulted in better packing of the tasks that require experts who are rare in the available technicians group. The FMM decreased the gap between best schedules and lower bounds of the problem instances in which rare expertness has been detected. Especially, our results are remarkably superior in cases of high number of technicians, tasks and skill domains besides the rare expertise.

The proposed matching model is a tool that provides a broad range of flexibility. This flexibility allows to construct a high number of schedules. As it can be seen in its mathematical formulation, weights of tasks are important parameters to obtain good quality solutions. There is no unique measurement for the weight of a task, but we proposed a measurement to include rare expertise concerns. We believe that our matching models are proper tools to test the success of these measurements. It will be useful to further explore the following points:

- decision on selection of teams and tasks to include in our matching models (e.g. randomization)
- developing new measurements or improving already proposed ones
- necessary modifications needed for adapting our matching model for the other versions of multi-skill workforce scheduling problems.

7 Acknowledgments

This research is supported by France Telecom/TUE Research agreement No. 46145963. We are grateful to Alexandre Laugier and Anne-Marie Bustos for their cooperation and help in testing various versions of the algorithm. Thanks are also due to Gerhard J. Woeginger for helpful discussions on the interpretation of MIP models.

References

- Ballou, D., Tayi, G., 1996, "A decision aid for the selection and scheduling of software maintenance projects", *IEEE Transactions on Systems Man and Cybernetics Part A Systems and Humans*, Vol 26, No.2, pp. 203-212.
- Bassett M., 2000, "Assigning projects to optimize the utilization of employees time and expertise", *Computers and Chemical Engineering*, Vol 24, pp. 1013-1021.
- Bellenguez M. O. and Neron E., 2007, "A Branch-and-bound method for solving multi-skill project scheduling problem", *RAIRO Operations Research*, Vol 41, pp. 155-170.
- Blazewicz, J., Lenstra, J.K. and Rinnooy Kan, A.H.G., 1983, "Scheduling projects subject to resource constraints: classification and complexity", *Discrete Applied Mathematics*, Vol 5, pp. 11-24.
- Brucker, P., Drexl, A., Möhring R., Neumann K., Pesch E., 1999, "Resource-constrained project scheduling: notation, classification, models, and methods", *European Journal of Operational Research*, Vol 112, No.1, pp. 3-41.
- Brucker, P., Knust, S., 2000, "A linear programming and constraint propagation-based lower bound for RCPSP", *European Journal of Operational Research*, Vol 127, pp. 355-362.
- Buddhakulsomsiri, J., Kim, D. S., 2006, "Properties of multi-mode resource-constrained project scheduling problems with resource vacations and activity splitting", *European Journal of Operational Research*, Vol 175, No.1, pp. 279-295.
- Cai, X., Li, K. N., 2000, "A genetic algorithm for scheduling staff of mixed skills under multi-criteria", *European Journal of Operational Research*, Vol.125 No.2, pp. 359-369.
- Cordeau, J. F., Laporte, G., Pasin F., Ropke, S., 2009, "Scheduling technicians and tasks in a telecommunication company", *Journal of Scheduling*, to appear.
- Demeulemeester, E. and Herroelen, W., 1992, "A branch-and-bound procedure for the multiple resource-constrained project scheduling problem", *Management Science*, Vol.38, pp. 1803-1818.
- Drezet, L-E. , Billaut, J-C. , 2008, "A project scheduling problem with labour constraints and time-dependent activities requirements", *International Journal of Production Economics*, Vol 112, pp. 217-225.
- Dutot, P., Laugier, A., Bustos, A., 2006, "Technicians and interventions scheduling for telecommunications", *Problem description of ROADEF 2007 Challenge*.
- Estellon, B., Gardi, F., Nouioua, K., 2009, "High-Performance local search for task scheduling with human resource allocation", *Lecture Notes In Computer Science*, Vol.5752 , pp. 1-15.
- Gutjahr, W. J.; Katzensteiner, S., Reiter, P., Stummer, C., Denk, M. , 2008, "Competence-driven project portfolio selection, scheduling and staff assignment", *Central European Journal of Operations Research*, Vol.16 No.3, pp. 281-306.

- 1
2
3
4
5
6
7
8 Hartmann, S., Briskorn, D., 2009, "A survey of variants and extensions of the resource-
9 constrained project scheduling problem", *Central European Journal of Operations Re-*
10 *search*, in press, corrected proof, DOI: 10.1016/j.ejor.2009.11.005.
11
- 12 Heilmann, R., 2003, "A branch-and-bound procedure for the multi-mode resource-constrained
13 project scheduling problem with minimum and maximum time lags", *European Journal of*
14 *Operational Research*, Vol.144 No.2, pp. 348-365.
15
- 16 Heilmann, R., Schwindt, C., 1997, "Lower bounds for RCPSP", *Technical report 511, WIOR,*
17 *Universitat Karlsruhe*, Karlsruhe, Germany.
18
- 19 Heimerl, C. and Kolisch, R., 2009, "Scheduling and staffing multiple projects with a multi-skilled
20 workforce", *OR Spectrum*, doi:10.1007/s00291-009-0169-4.
21
- 22 Hurkens, C.A.J., 2009, "Incorporating the strength of MIP modeling in schedule construction",
23 *RAIRO Operations Research*, Vol. 43, pp. 409-420.
24
- 25 Li, H. and Womer K., 2009, "Scheduling projects with multi-skilled personnel by a hybrid
26 MILP/CP benders decomposition algorithm", *Journal of Scheduling*, Vol.12 No.3, pp. 281-
27 298.
28
- 29 Mingozi, A., Maniezzo, V., Ricciardelli, S. and Bianco, L. , 1998, "An exact algorithm for the
30 multiple resource-constrained project scheduling problem based on a new mathematical
31 formulation", *Management Science*, Vol.44, pp. 714-729.
32
- 33 Möhring, R.H., Schulz, A. S., Stork, F., Uetz, M. , 2003, "Solving project scheduling problems
34 by minimum cut computations", *Management Science*, Vol.49, No.3, pp. 330-350.
35
- 36 Reyck, B. and Herroelen, W., 1999, "The multi-mode resource-constrained project scheduling
37 problem with generalized precedence relations", *European Journal of Operational Research*,
38 Vol.119 No.2, pp.538-556.
39
- 40 Valls, V., Perez, A., Quintanilla, S., 2009, "Skill workforce scheduling in service centers", *Eu-*
41 *ropean Journal of Operational Research*, Vol 193, No.3, pp. 791-804.
42
- 43 Wu, M.C. and Sun, S.H., 2006, "A project scheduling and staff assignment model considering
44 learning effect", *The International Journal of Advanced Manufacturing Technology*, Vol 28,
45 pp. 1190-1195.
46
- 47 Yoshimura, M., Fujimi, Y., Izui, K., Nishiwaki, S., 2006, "Decision-making support system for
48 human resource allocation in product development projects ", *International Journal of*
49 *Production Research*, Vol 44, No. 5, pp. 831-848.
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

Working Papers Beta 2009 - 2010

nr.	Year	Title	Author(s)
327	2010	A combinatorial approach to multi-skill workforce scheduling	Murat Firat, Cor Hurkens
326	2010	Stability in multi-skill workforce scheduling	Murat Firat, Cor Hurkens, Alexandre Laugier
325	2010	Maintenance spare parts planning and control: A framework for control and agenda for future research	M.A. Driessen, J.J. Arts, G.J. v. Houtum, W.D. Rustenburg, B. Huisman
324	2010	Near-optimal heuristics to set base stock levels in a two-echelon distribution network	R.J.I. Basten, G.J. van Houtum
323	2010	Inventory reduction in spare part networks by selective throughput time reduction	M.C. van der Heijden, E.M. Alvarez, J.M.J. Schutten
322	2010	The selective use of emergency shipments for service-contract differentiation	E.M. Alvarez, M.C. van der Heijden, W.H. Zijm
321	2010	Heuristics for Multi-Item Two-Echelon Spare Parts Inventory Control Problem with Batch Ordering in the Central Warehouse	B. Walrave, K. v. Oorschot, A.G.L. Romme
320	2010	Preventing or escaping the suppression mechanism: intervention conditions	Nico Dellaert, Jully Jeunet.
319	2010	Hospital admission planning to optimize major resources utilization under uncertainty	R. Seguel, R. Eshuis, P. Grefen.
318	2010	Minimal Protocol Adaptors for Interacting Services Teaching Retail Operations in Business and Engineering Schools	Tom Van Woensel, Marshall L. Fisher, Jan C. Fransoo.
317	2010	Design for Availability: Creating Value for Manufacturers and Customers	Lydie P.M. Smets, Geert-Jan van Houtum, Fred Langerak.
316	2010	Transforming Process Models: executable rewrite rules versus a formalized Java program	Pieter van Gorp, Rik Eshuis.
315	2010		Bob Walrave, Kim E. van Oorschot, A. Georges L. Romme
314	2010	Ambidexterity and getting trapped in the suppression of exploration: a simulation model	

313	2010	A Dynamic Programming Approach to Multi-Objective Time-Dependent Capacitated Single Vehicle Routing Problems with Time Windows	S. Dabia, T. van Woensel, A.G. de Kok
312	2010	Tales of a So(u)rcerer: Optimal Sourcing Decisions Under Alternative Capacitated Suppliers and General Cost Structures	Osman Alp, Tarkan Tan
311	2010	In-store replenishment procedures for perishable inventory in a retail environment with handling costs and storage constraints	R.A.C.M. Broekmeulen, C.H.M. Bakx
310	2010	The state of the art of innovation-driven business models in the financial services industry	E. Lüftenegger, S. Angelov, E. van der Linden, P. Grefen
309	2010	Design of Complex Architectures Using a Three Dimension Approach: the CrossWork Case	R. Seguel, P. Grefen, R. Eshuis
308	2010	Effect of carbon emission regulations on transport mode selection in supply chains	K.M.R. Hoen, T. Tan, J.C. Fransoo, G.J. van Houtum
307	2010	Interaction between intelligent agent strategies for real-time transportation planning	Martijn Mes, Matthieu van der Heijden, Peter Schuur
306	2010	Internal Slackening Scoring Methods	Marco Slikker, Peter Borm, René van den Brink
305	2010	Vehicle Routing with Traffic Congestion and Drivers' Driving and Working Rules	A.L. Kok, E.W. Hans, J.M.J. Schutten, W.H.M. Zijm
304	2010	Practical extensions to the level of repair analysis	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
303	2010	Ocean Container Transport: An Underestimated and Critical Link in Global Supply Chain Performance	Jan C. Fransoo, Chung-Yee Lee
302	2010	Capacity reservation and utilization for a manufacturer with uncertain capacity and demand	Y. Boulaksil; J.C. Fransoo; T. Tan
300	2009	Spare parts inventory pooling games	F.J.P. Karsten; M. Slikker; G.J. van Houtum
299	2009	Capacity flexibility allocation in an outsourced supply chain with reservation	Y. Boulaksil, M. Grunow, J.C. Fransoo
298	2010	An optimal approach for the joint problem of level of repair analysis and spare parts stocking	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
297	2009	Responding to the Lehman Wave: Sales Forecasting and Supply Management during the Credit Crisis	Robert Peels, Maximiliano Udenio, Jan C. Fransoo, Marcel Wolfs, Tom Hendrikx
296	2009	An exact approach for relating recovering surgical patient workload to the master surgical schedule	Peter T. Vanberkel, Richard J. Boucherie, Erwin W. Hans, Johann L. Hurink, Wineke A.M. van Lent, Wim H. van Harten
295	2009	An iterative method for the simultaneous optimization of repair decisions and spare parts stocks	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten

294	2009	Fujaba hits the Wall(-e)	Pieter van Gorp, Ruben Jubeh, Bernhard Grusie, Anne Keller
293	2009	Implementation of a Healthcare Process in Four Different Workflow Systems	R.S. Mans, W.M.P. van der Aalst, N.C. Russell, P.J.M. Bakker
292	2009	Business Process Model Repositories - Framework and Survey	Zhiqiang Yan, Remco Dijkman, Paul Grefen
291	2009	Efficient Optimization of the Dual-Index Policy Using Markov Chains	Joachim Arts, Marcel van Vuuren, Gudrun Kiesmuller
290	2009	Hierarchical Knowledge-Gradient for Sequential Sampling	Martijn R.K. Mes; Warren B. Powell; Peter I. Frazier
289	2009	Analyzing combined vehicle routing and break scheduling from a distributed decision making perspective	C.M. Meyer; A.L. Kok; H. Kopfer; J.M.J. Schutten
288	2009	Anticipation of lead time performance in Supply Chain Operations Planning	Michiel Jansen; Ton G. de Kok; Jan C. Fransoo
287	2009	Inventory Models with Lateral Transshipments: A Review	Colin Paterson; Gudrun Kiesmuller; Ruud Teunter; Kevin Glazebrook
286	2009	Efficiency evaluation for pooling resources in health care	P.T. Vanberkel; R.J. Boucherie; E.W. Hans; J.L. Hurink; N. Litvak
285	2009	A Survey of Health Care Models that Encompass Multiple Departments	P.T. Vanberkel; R.J. Boucherie; E.W. Hans; J.L. Hurink; N. Litvak
284	2009	Supporting Process Control in Business Collaborations	S. Angelov; K. Vidyasankar; J. Vonk; P. Grefen
283	2009	Inventory Control with Partial Batch Ordering	O. Alp; W.T. Huh; T. Tan
282	2009	Translating Safe Petri Nets to Statecharts in a Structure-Preserving Way	R. Eshuis
281	2009	The link between product data model and process model	J.J.C.L. Vogelaar; H.A. Reijers
280	2009	Inventory planning for spare parts networks with delivery time requirements	I.C. Reijnen; T. Tan; G.J. van Houtum
279	2009	Co-Evolution of Demand and Supply under Competition	B. Vermeulen; A.G. de Kok
278	2010	Toward Meso-level Product-Market Network Indices for Strategic Product Selection and (Re)Design Guidelines over the Product Life-Cycle	B. Vermeulen, A.G. de Kok
277	2009	An Efficient Method to Construct Minimal Protocol Adaptors	R. Seguel, R. Eshuis, P. Grefen
276	2009	Coordinating Supply Chains: a Bilevel Programming Approach	Ton G. de Kok, Gabriella Muratore
275	2009	Inventory redistribution for fashion products under demand parameter update	G.P. Kiesmuller, S. Minner
274	2009	Comparing Markov chains: Combining aggregation and precedence relations applied to sets of states	A. Busic, I.M.H. Vliegen, A. Scheller-Wolf

273	2009	Separate tools or tool kits: an exploratory study of engineers' preferences	I.M.H. Vliegen, P.A.M. Kleingeld, G.J. van Houtum
272	2009	An Exact Solution Procedure for Multi-Item Two-Echelon Spare Parts Inventory Control Problem with Batch Ordering	Engin Topan, Z. Pelin Bayindir, Tarkan Tan
271	2009	Distributed Decision Making in Combined Vehicle Routing and Break Scheduling	C.M. Meyer, H. Kopfer, A.L. Kok, M. Schutten
270	2009	Dynamic Programming Algorithm for the Vehicle Routing Problem with Time Windows and EC Social Legislation	A.L. Kok, C.M. Meyer, H. Kopfer, J.M.J. Schutten
269	2009	Similarity of Business Process Models: Metrics and Evaluation	Remco Dijkman, Marlon Dumas, Boudewijn van Dongen, Reina Kaarik, Jan Mendling
267	2009	Vehicle routing under time-dependent travel times: the impact of congestion avoidance	A.L. Kok, E.W. Hans, J.M.J. Schutten
266	2009	Restricted dynamic programming: a flexible framework for solving realistic VRPs	J. Gromicho; J.J. van Hoorn; A.L. Kok; J.M.J. Schutten;

Working Papers published before 2009 see: <http://beta.ieis.tue.nl>