EMPLOYEE / SCHOOL / SPORTS TIMETABLING: WHAT A TRIPLET!

F. Della Croce
D.A.I. Politecnico di Torino, Italy
(currently visiting LAMSADE / Univ. Paris Dauphine)

OUTLINE

- Introduction
- Employee Timetabling
- School Timetabling
- Sport Timetabling
- A couple of applications
  - Scheduling the Italian Football League: an ILP based approach.
  - A greedy-based neighborhood search approach to a nurse rostering problem.

Introduction

- Timetabling (definitions)
  - Wren (1996):
    "Timetabling is the problem of placing certain resources, subject to constraints, into a limited number of time slots and places with the aim being to satisfy a set of stated objectives to the highest possible extent".
  - Burke-Petrovic (2004):
    "Timetabling can be considered to be a certain type of scheduling problem".
- Timetabling problems arise in a wide variety of domains
  - Employee timetabling:
    Healthcare institutions (nurse and surgeon rostering), transport (train and bus timetabling, trains and planes crew scheduling), call centers.
  - School timetabling:
    High school timetabling, University timetabling (courses and exams).
  - Sport timetabling:
    round robin tournaments.

Literature

- "The Practice and Theory of Automated Timetabling I-II-III-IV-V: Selected Papers from the PATAT Conferences (International Conference on the Practice and Theory of Automated Timetabling)", Springer LNCS Series 1996 - 1998 - 2001 - 2003 - 2005 (forthcoming).
- Special issue of CAOR "Operations research in sport", forthcoming (partially available on Science Direct - articles in press).
- Special Issue of AOR on "Staff Scheduling and Rostering: Theory and Applications", Parts I and II, Volumes 127-128, 2004 (with a comprehensive annotated bibliography reviewing over 700 papers!).
- Feature issue of EJOR on "Timetabling and Rostering" Volume 153, Issue 1, 2004.
- "Handbook of Scheduling: Algorithms, Models, and Performance Analysis", CRC Press, 2004.
  Chapter 44 (Nurse rostering), chapter 45 (University timetabling), chapter 52 (sports scheduling).
- ...

Conferences, Working groups, bulletins

- PATAT Conference (International Conference on the Practice and Theory of Automated Timetabling) - bi-annual every even year.
- WATT: the EURO Working Group on Automated Timetabling ("*formed to discuss, promote, and perform research into automated timetabling issues and methods*").
  Organizes WATT workshops at EURO-Conferences typically every odd year.
  See website http://grumpy.cs.nott.ac.uk/ASAP/watt/.
- Bulletin: WATT digest (first issue december 1996) produced approximately at quarterly intervals from spring 2003.

Approaches

- Mathematical programming
  - ILP modeling and decomposition.
  - Specific CO techniques (e.g. set covering / set partitioning models plus column generation approaches).
- Metaheuristics
  - All sorts of neighborhood search approaches.
  - However: often no natural neighborhoods available.
- Constraint programming
  - Mostly applied to sports timetabling.
  - Best results (as usual) when there is "no" objective function (just search of a feasible solution).
  - ...

Employee timetabling

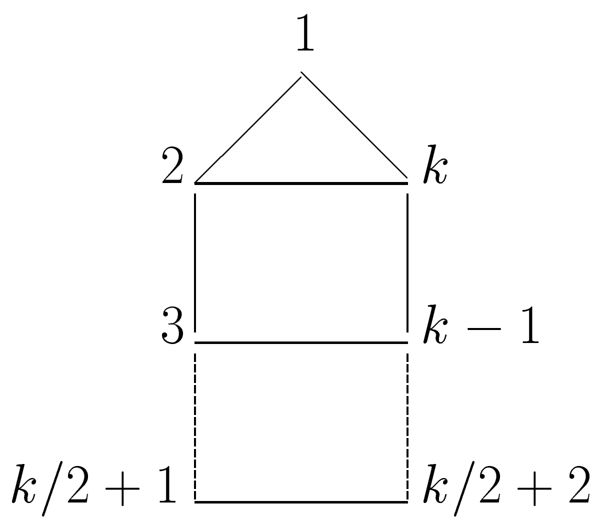
- Mostly problem oriented approaches.
- A recent successful application: train scheduling and rostering of the italian railways (Caprara et al. 1997-2001) with the solution method classified (in the annotated bibliography) as a mixture of *"Integer programming, Lagrangian relaxation, Set covering, constructive heuristic"*.
- No general purpose strong structural properties (e.g., distinct nurse rostering applications require in some cases and do not imply in some other cases weekly patterns).
- Correspondingly no strong theoretical results strictly related to employee timetabling.

School timetabling

- Much more structured problems.
- Still quite a difference between University and High School Timetabling.
  - University timetabling:
    needs to match courses, halls and lecturers but it is not subject to hard "temporal constraints".
  - High School Timetabling:
    needs to match just courses to lecturers but it is subject to hard "temporal constraints" (no "holes" in the timetable).
- Most applications are typically based on ILP modeling and decomposition.
- No strong theoretical results.
- Possibly room for neighborhood search approaches based on exponential size neighborhoods explored in polynomial time (the so-called VLS neighborhood appraoches)

Sports timetabling

- The most structured problems of the triplet.
- Round robin tournaments (RRT): both single (SRRT) and double mirrored (DMRRT).
- Strong links with graph theory: an SRRT with an even number k of teams corresponds to the one-factorization of a complete graph with k vertices.
- Strong links with latin squares: a k X k symmetric latin square with the restriction that the elements down the principle diagonal are all identical generates an SRRT with an even number k of teams.
- computing an SRRT is easy: the so-called circle-design method.



Home/away patterns notation

(example on a 8-Team Tournament)

Pattern: string of symbols HA indicating the sequence of Home (H) and Away (A) matches of a team in the tournament

H A A H A H A

Break: two consecutive matches played home or away

H **A** **A** H A H A

Pattern Set: A set of patterns having cardinality equal to the # of teams in the tournament

1: H A A H A H A
2: A H H A H A H
3: H A H A H A H
4: A H A H A H A
5: H A H H A H A
6: A H A A H A H
7: H A H A A H A
8: A H A H H A H

Round robin tournaments with home-away patterns requirements

- Computing an SRRT or a DMRRT minimizing the number of breaks is easy.
- With an even number of teams, there are at least $n-2$ breaks for SRRT and $3n-6$ breaks for DMRRT.
- Starting from the circle-design solution, it is always possible to assign home and away matches with exactly $n-2$ breaks for SRRT and $3n-6$ breaks for DMRRT.
- However computing RRTs minimizing the number of breaks plus various other practical constraints is hard.

APPLICATIONS

- Scheduling the Italian Football League: an ILP based approach.
- A greedy-based neighborhood search approach to a nurse rostering problem.

Italian Football League problem characteristics

The Italian Football League tournament calendar presents the following characteristics

- Round Robin Tournament
- Home/Away matches
- Seeded Teams / Teams located in the same town (Derbies)
- First and Last $\gamma = 3$ weeks of the tournament without matches between Seeded Teams or Derbies
- TV coverage of the matches by two concurrent cable TVs

Objectives

- # breaks minimization
- Balanced TV coverage
- Handling a predefined # Seeded Teams

Solution approach

- Literature: Solution procedures typically based on a 3-phase approach
  (ref. work - TN98 - on the ACC basketball Conference with 9 teams)
  1. Generation of all pattern sets
  2. Generation of all possible calendars with the above pattern sets
  3. Exhaustive generation of feasible [patterns / team] assignments
- We tested two approaches:
  1. matching pattern sets to predefined calendars
  2. apply the 3-phase approach to the considered problem
- and obtained the following results
  1. it is hard to get feasible schedules
  2. it is a viable approach provided that appropriate adjustment are devised

3-phase approach

1. Need to handle the combinatorial explosion of pattern sets and calendars in phase 1 (and 2)

    - iteratively generates a limited (but sufficient) # of feasible pattern sets by excluding solutions that are too similar (phase 1).
    - search for one calendar for each feasible pattern set (phase 2)

2. Cannot generate all possible assignments of reals teams to pattern sets (phase 3)

    - final assignment by means of an ILP-based approach.

3. Indeed all phases can be successfully handled via ILP models optimally solved by commercial softwares.

3-phase approach

Given all feasible patterns (no more than 4 breaks are allowed), the method iteratively proceeds as follows:

1. Determine a balanced pattern set with respect to TV coverage minimizing # of breaks and presenting a sufficient # of complementary patterns (teams located in the same town).

2. Check if a feasible timetable can be obtained from the considered pattern set.

3. Assign the real teams to the timetable patterns so that the constraints on derbies and seeded teams are satisfied. Further specific constraints on the real teams can be handled in this phase.

By generating different feasible pattern sets, it is possible to obtain different final calendars.

First phase: ILP model

- Cost function: # of breaks minimization
- Constraints:
  1. # of Home matches assigned to each TV (TV coverage)
  2. # of patterns assigned to each TV
  3. Complementary patterns handling
  4. Each pattern can be selected just once
- To get further different patterns, it is sufficient to add a constraint forbidding in the next solution to have more than 50% of the patterns selected in the current solution.
- With 18 teams, there are ≈ 200 0/1 variables and ≈ 500 constraints.

Second phase: ILP model

- No (hence, fictituous) cost function

- Constraints:

  1. for each week and Home (Away) pattern, select exactly one Away (Home) opponent pattern
  2. if pattern j is the opponent of pattern i on week t, then pattern i is the opponent of pattern j on week t
  3. each pair of patterns i, j is selected for just one week
  4. each team (pattern) plays just one match each week

- With 18 teams there are ≈ 5000 0/1 variables and ≈ 10000 constraints

Third phase: ILP model

- Cost function: maximize a general function related to additional requirements of the teams

- Constraints:

  1. each pattern is matched to exactly one team
  2. each team is matched to exactly one pattern
  3. matches between seeded teams cannot occur in the first and last γ matches
  4. derbies cannot occur in the first and last γ matches
  5. teams of the same town require complementary patterns

- With 18 teams there are $\approx$ 200 0/1 variables and $\approx$ 200 constraints.

A small illustrative example

- Six teams A, B, C, D, E, F are considered with A, B, C, D belonging to TV 1 and E, F belonging to TV 2.
- There are two seeded teams A, E and two teams located in the same town A, B.
- The matches A vs E and A vs B cannot occur in weeks 1 and 5 (first and last week of the tournament).

Below are all the patterns considered in phase 1 with their corresponding coefficient in the cost function.

INDEX	PATTERN	OBJECTIVE FUNCTION COEFFICIENT
P_1	$AHAHA$	0
P_2	$AHHAH$	3
P_3	$AHAAH$	3
P_4	$HAHAH$	0
P_5	$HAAHA$	3
P_6	$HAHHA$	3
P_7	$AAHAA$	4
P_8	$AAHHA$	4
P_9	$AHHAA$	4
P_{10}	$HHAHH$	4
P_{11}	$HHAAH$	4
P_{12}	$HAAHH$	4

A small illustrative example

- The first ILP model has 24 binary variables x_{ik} where x_{ik} is equal to 1 if pattern P_i is assigned to TV k.
- The bound $3n-6$ on the number of breaks is equal to 12.
- The first pattern set has $[P_2, P_3, P_5, P_6]$ assigned to TV 1 and $[P_1, P_4]$ assigned to TV 2 with cost function value $= 12$.
- With this pattern set, phase 2 outputs the feasible calendar depicted below (the first pattern indicated in each column plays home)

WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5
P_4-P_2	P_1-P_6	P_2-P_3	P_1-P_2	P_2-P_6
P_5-P_1	P_2-P_5	P_4-P_1	P_5-P_3	P_3-P_1
P_6-P_3	P_3-P_4	P_6-P_5	P_6-P_4	P_4-P_5

A small illustrative example

- Phase 3 matches patterns and teams as follows

$$[A - P_3, \quad B - P_2, \quad C - P_5, \quad D - P_6]$$
$$[E - P_4, \quad F - P_1]$$

- The first generated calendar is

WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5
C-F	A-E	B-A	C-A	A-F
D-A	B-C	D-C	D-E	B-D
E-B	F-D	E-F	F-B	E-C

- Exactly two teams belonging to TV 1 (A, B, C, D) and one team belonging to TV 2 (E, F) are scheduled to play home on each week.
- Match $A - E$ between seeded teams is scheduled on week 2.
- Match between teams A, B belonging to the same town is scheduled on week 3.

Computational testing

- The procedure was tested on a Pentium IV 1500 with 256 Mb RAM applying LINGO 7.0 as ILP solver.
- For 18-team instances, the first solution was achieved in a couple of minutes and the first five solutions in less than 15 minutes.
- On the average less than 100 pattern sets were necessary to reach the first five feasible calendars.
- We considered the real data of the Serie A for the years $2001-2002$, $2002-2003$ and $2003-2004$.
- For all cases we have 4 seeded teams on a total of 18 teams (min. # of breaks =48).

Year	Official calendar		Proposed solution (avg)	
	breaks	TV coverage violations	breaks	TV coverage violations
2001/2002	58	14	50	0
2002/2003	58	4	49.6	0
2003/2004	60	12	50.8	0

The official 2003/2004 calendar of the Italian Serie A

WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6
ANC-MIL	*CHI*-JUV	*ANC*-*MOD*	*CHI*-*PER*	*ANC*-UDI	*ANC*-JUV
BOL-PAR	*EMP*-REG	BOL-UDI	*EMP*-LAZ	*EMP*-*MOD*	BOL-*PER*
BRE-*CHI*	LEC-*ANC*	*BRE*-REG	MIL-LEC	**INT-MIL**	*BRE*-INT
INT-*MOD*	MIL-BOL	INT-SAM	*MOD*-BOL	JUV-BOL	*CHI*-SAM
JUV-*EMP*	*MOD*-UDI	JUV-ROM	PAR-SIE	LAZ-*CHI*	**MIL-LAZ**
LAZ-LEC	PAR-*PER*	LAZ-PAR	REG-JUV	LEC-*BRE*	*MOD*-LEC
PER-SIE	ROM-*BRE*	LEC-*CHI*	ROM-*ANC*	PAR-SAM	REG-SIE
REG-SAM	SAM-LAZ	*PER*-MIL	SAM-*BRE*	*PER*-REG	ROM-PAR
UDI-ROM	SIE-INT	SIE-*EMP*	UDI-INT	SIE-ROM	UDI-*EMP*

WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12
EMP-*CHI*	*ANC*-SIE	*BRE*-BOL	*ANC*-*BRE*	*BRE*-UDI	BOL-*ANC*
INT-ROM	BOL-SAM	INT-*ANC*	BOL-ROM	**JUV-INT**	*CHI*-ROM
JUV-*BRE*	*BRE*-PAR	JUV-UDI	*CHI*-MIL	MIL-*MOD*	*EMP*-MIL
LAZ-BOL	*CHI*-INT	PAR-MIL	*EMP*-PAR	PAR-*CHI*	INT-*PER*
PAR-*MOD*	LEC-*EMP*	*PER*-LEC	INT-REG	*PER*-*EMP*	**LAZ-JUV**
PER-UDI	**MIL-JUV**	REG-*MOD*	LAZ-*PER*	REG-BOL	LEC-PAR
REG-*ANC*	*MOD*-*PER*	**ROM-LAZ**	LEC-SAM	ROM-LEC	*MOD*-*BRE*
SAM-MIL	ROM-REG	SAM-*EMP*	*MOD*-JUV	SAM-*ANC*	SAM-SIE
SIE-LEC	UDI-LAZ	SIE-*CHI*	UDI-SIE	SIE-LAZ	UDI-REG

WEEK 13	WEEK 14	WEEK 15	WEEK 16	WEEK 17
ANC-LAZ	*CHI*-*ANC*	*ANC*-PAR	*CHI*-UDI	*ANC*-*PER*
BOL-INT	*EMP*-ROM	BOL-*EMP*	*EMP*-*ANC*	BOL-*CHI*
BRE-*EMP*	**LAZ-INT**	*BRE*-SIE	LAZ-*BRE*	*BRE*-MIL
JUV-PAR	LEC-JUV	INT-LEC	LEC-BOL	INT-*EMP*
MIL-SIE	MIL-UDI	JUV-*PER*	MIL-REG	JUV-SIE
PER-SAM	PAR-REG	*MOD*-*CHI*	PAR-INT	*MOD*-LAZ
REG-*CHI*	*PER*-*BRE*	REG-LAZ	*PER*-ROM	REG-LEC
ROM-*MOD*	SAM-*MOD*	ROM-MIL	SAM-JUV	ROM-SAM
UDI-LEC	SIE-BOL	UDI-SAM	SIE-*MOD*	UDI-PAR

One of the calendars generated for the 2003/2004 tournament

WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6
ANC-*MOD*	BOL-SAM	*BRE*-REG	*ANC*-SAM	*BRE*-LAZ	*ANC*-PAR
CHI-REG	*BRE*-*CHI*	*CHI*-*EMP*	BOL-UDI	*CHI*-JUV	BOL-*CHI*
MIL-*BRE*	*EMP*-PAR	MIL-BOL	*EMP*-*BRE*	MIL-*MOD*	**INT-MIL**
PAR-LEC	INT-*PER*	PAR-LAZ	INT-SIE	PAR-INT	JUV-*BRE*
PER-BOL	JUV-UDI	*PER*-*ANC*	JUV-PAR	*PER*-SAM	LAZ-*EMP*
ROM-*EMP*	LAZ-*ANC*	ROM-JUV	LAZ-*CHI*	REG-*EMP*	LEC-REG
SAM-INT	LEC-MIL	SAM-*MOD*	LEC-ROM	ROM-*ANC*	*MOD*-SIE
SIE-JUV	*MOD*-ROM	SIE-LEC	*MOD*-*PER*	SIE-BOL	*PER*-UDI
UDI-LAZ	REG-SIE	UDI-INT	REG-MIL	UDI-LEC	SAM-ROM

WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12
BRE-BOL	*ANC*-SIE	*BRE*-INT	*ANC*-MIL	BOL-*ANC*	*ANC*-*BRE*
CHI-SAM	BOL-REG	*CHI*-*ANC*	BOL-*EMP*	*BRE*-SAM	**INT-JUV**
EMP-LEC	INT-*CHI*	*EMP*-*MOD*	INT-REG	*CHI*-PAR	*MOD*-BOL
MIL-JUV	JUV-*EMP*	LAZ-LEC	**JUV-LAZ**	*EMP*-*PER*	PAR-REG
PAR-*MOD*	**LAZ-MIL**	MIL-*PER*	LEC-*CHI*	**LAZ-INT**	*PER*-LEC
REG-LAZ	LEC-*BRE*	PAR-BOL	*MOD*-*BRE*	LEC-JUV	**ROM-LAZ**
ROM-INT	*MOD*-UDI	REG-JUV	*PER*-PAR	MIL-ROM	SAM-MIL
SIE-*PER*	*PER*-ROM	SIE-ROM	ROM-UDI	REG-*MOD*	SIE-*CHI*
UDI-*ANC*	SAM-PAR	UDI-SAM	SAM-SIE	SIE-UDI	UDI-*EMP*

WEEK 13	WEEK 14	WEEK 15	WEEK 16	WEEK 17
BOL-INT	*ANC*-JUV	*BRE*-*PER*	*ANC*-*EMP*	*BRE*-UDI
BRE-ROM	INT-*EMP*	*CHI*-*MOD*	BOL-LAZ	*CHI*-ROM
CHI-*PER*	*MOD*-LEC	*EMP*-SAM	MIL-*CHI*	*EMP*-MIL
EMP-SIE	PAR-*BRE*	JUV-BOL	*MOD*-INT	INT-*ANC*
JUV-SAM	*PER*-REG	LAZ-SIE	*PER*-JUV	JUV-*MOD*
LAZ-*MOD*	ROM-BOL	LEC-INT	ROM-REG	LAZ-*PER*
LEC-*ANC*	SAM-LAZ	MIL-UDI	SAM-LEC	LEC-BOL
MIL-PAR	SIE-MIL	PAR-ROM	SIE-*BRE*	PAR-SIE
REG-UDI	UDI-*CHI*	REG-*ANC*	UDI-PAR	REG-SAM

Nurse rostering problem characteristics

- Planning nurses' monthly shifts
- Every day a shift or a day-off must be assigned to each nurse
- Three shifts a day: morning, afternoon and night shifts
- Various types of requirements:
  - Contractual requirements
  - Operational requirements
  - Other requirements

Contractual requirements

C1) The number of days-off per month must be equal to a predefined value, provided by the ward direction;

C2) The holidays chosen by the nurses are compulsory;

C3) a nurse cannot work for more than a predefined value of consecutive days;

C4) night shifts must be allocated in sets of minimum/ maximum consecutive days;

C5) after a set of night shifts, a nurse must have at least a predefined number of days-off;

C6) after a set of night shifts, a nurse cannot be assigned again to a night shift before at least a predefined number of days.

Operational requirements

O1) In each shift, a minimum number of nurses must be guaranteed. This value number may differ from shift to shift and from day to day;

O2) working shifts must be evenly assigned;

O3) working shifts and day-off in the week-end should be evenly allocated;

O4) afternoon shift/morning shift sequences must be avoided;

O5) nurses' requirements must be satisfied as much as possible.

Other requirements

- It is recommended to assign a set of morning shifts before the first day of a period of holidays and a set of night shifts after a period of holidays;
- it is recommended to assign two days-off when the maximum number of consecutive working days has been reached;
- it is recommended not to assign a night shift before a requested day-off.

Modelling the problem

Contractual and operational requirements are partially in conflict $\Longrightarrow$ only a part of them are considered as problem constraints:

- contractual requests C2-C6 become constraints of the model;
- contractual request C1 and all the operational requests become objectives;
- goals:
  - to guarantee that the number of days-off is equal to a fixed value;
  - to minimize covering violations requirements and sequences afternoon shift/morning shift;
  - to assign evenly shifts and day-offs;
  - to meet nurses' other requirements as much as possible.

Objective function

$$\min z = \alpha \cdot A + \beta \cdot B + \gamma \cdot C + \delta \cdot D + \varepsilon \cdot E$$

$$\alpha \gg \beta \gg \gamma \gg \delta \gg \epsilon$$

- A is the difference between the total amount of days-off predefined by the ward direction and the corresponding amount of days-off assigned by the algorithm;
- B is the total number of covering violations on night shifts;
- C is the total number of multiple covering violations on daily shifts;
- D is the total number of single covering violations on daily shifts;
- E is a linear combination of different factors (number of afternoon shift/morning shift sequences, even assignment of shifts, respect of nurses' requirements).

Neighborhood search solution approach

- Initial solution
  - greedy algorithm

- Neighborhood search (TS/ILS)
  - solution representation
  - neighborhood

Initial solution

1. Holidays and requested days-off assignment
2. Night shifts assignment
3. Morning and afternoon shifts assignment

⇒ For every day and for each type of shift the best candidate to work in that shift is selected according to the increasing number of shifts of that type already assigned to that candidate.

Neighborhood search

- Solution representation
  - A complete solution defines for each day of the month a shift for each nurse;
  - moves which operate on every kind of shifts can easily lead to unfeasible solutions;
  - we operate on partial solutions represented by holidays, days-off requested by nurses, night shifts and days-off linked to the night shifts;
  - a partial neighbor solution is then completed by means of the greedy algorithm.
- Neighborhood
  - I. A new set of night shifts is moved from a nurse to another;
  - II. the first night of a set is moved from one nurse to another;
  - III. the last night of a set is moved from one nurse to another;
  - IV. one night shift is added to a nurse as first or last night shift of a set.

Diversification - Multistart

In order to obtain good initial solutions, we embed the greedy algorithm in three multistart procedures:

- the first multistart procedure tries to fulfil the requested night coverage in as many days as possible;
- the second multistart procedure tries to find a solution in which every nurse has the correct number of days-off by modifying the number of night shifts assigned to each nurse;
- the third multistart procedure tries to find a solution with a better objective function by varying the priorities according to whom the best candidate for each shift is chosen.

Computational testing

- The procedure was tested on several real life instances with ≈ 20 nurses on a Pentium IV 2 GHz:
  - on each instance, the greedy solution already improves upon the one proposed by the ward direction;
  - the greedy solution is considerably improved by the neighborhood search approach;
  - the overall procedure reaches in general the optimal values of factors A,B,C and D (by comparing the results obtained with trivially computable lower bounds).
  - CPU time strictly limited: less than 6 minutes for the overall procedure.
- The procedure was tested also on randomly generated instances with different problem sizes: the CPU time (single start) remains limited also for medium-large size problems (less than 5 hours on the average with 60 nurses).
- the proposed software is currently used in the hospital ward.