Addendum to :
Efficient Algorithms
for Automated Staff Scheduling

Andreas Fest* Berit Johannes Carola Schaad Marc Uetz

Technische Universität Berlin
Institut für Mathematik, MA 6–1
Straße des 17. Juni 136
10623 Berlin, Germany

January 2002

Abstract

This is an addendum to the final report on a cooperation between the ATOSS Software AG in Munich and the TU Berlin. The goal of this project was the extension of the staff scheduling algorithm described in [1] to promote block structures in the resulting schedule. A schedule has block structure if for each staff member the number of consecutive days with same characteristics (e.g. with the same shift) lies between a given minimum and a given maximum. In this report we give a description of the new feature of the algorithm and provide some test results based upon two adapted problem instances from ATOSS Software AG.

*Correspondence to: `fest@math.tu-berlin.de`

Contents

1 Introduction

This document is an extension of the final report [1] of the cooperation between the ATOSS Software AG and the Technische Universität Berlin. The purpose of this project was the integration of new cost components into the staff scheduling algorithm described in [1] to promote block structures in the computed staff schedule. We refer to [1] for a detailed description of the original problem and the implemented staff scheduling algorithm. The new feature is as follows. Each shift belongs to a certain shift group. There are two numbers associated with each shift group. The first number, further referred to as *min_cons_shift*, represents the minimal required number of consecutive days an employee has to be assigned to a shift of the same shift group. The second number, further referred to as *max_cons_shift*, represents the maximal allowed number of consecutive days an employee may be assigned to a shift of the same shift group. Thus, whenever an employee is assigned to a shift of a certain shift group, he/she has to be assigned to shifts of the same shift group on at least min_cons_shift consecutive days and at most max_ cons_ shift consecutive days. Furthermore, the number of consecutive working days and the number of consecutive days of absence are bounded from below and above by numbers given in the input. The "Zusatz zu Anlage A" of the FuE contract gives a detailed description of the modified problem.

The implemented algorithm consists of two phases, a *construction heuristic* that computes a starting solution, and a *local search* procedure, that tries to generate a feasible schedule with minimal total cost. In this report we first describe the change of the cost functions to accommodate the new feature. Then we give some hints how to set the parameters in the objective function to find a proper balance for the trade off between a block structure of the schedule and the other constraints. These recommendations are based on some test results obtained by using adjusted instances received from ATOSS Software AG.

2 Theoretical Hardness of the Problem

Of course, the considered staff scheduling problem is not easier to solve with the added constraints. The problem still belongs to the class of *NP-hard* optimization problems and even the problem to find a feasible solution is generally *NP-hard*. In particular, since the new block constraints might contradict some of the other constraints, for instance the constraints associated with wishes, we have to expect our solutions to now violate even more constraints. By tuning the parameters in the objective function the violations can be shifted towards the wish fulfillment constraints (or similar) or to the block constraints.

However, the algorithm still computes a staff schedule that

- always satisfies the *double booking constraint*,
- and which does not violate the given *fixed assignments*,

unless the input is obviously infeasible.

3 Integration of the New Constraints into the Solution Procedure

To promote block structures in the outcoming schedule, several new components of the objective function have to be considered. This section gives an overview of the additional components and how they are integrated into the solution procedure.

The solution procedure described in [1] consists of two main phases, the *construction heuristic procedure* and the *local search procedure*. The first phase computes a heuristic start solution by means of network flow techniques. The costs of total excess or total shortfall are considered during the heuristic procedure by augmenting the costs of arcs in the Min-Cost-Flow-Graph that would imply a violation of the new constraints or would make an existing violation worse.

The second phase is a local search algorithm that aims at improving the start solution in terms of the given objective function and consideres both the total and the maximal excess, respective shortfall of block length.

3.1 The New Constraints

The following new constraints are integrated into the algorithm to promote block structures in the schedule. We handle these new criteria in the same way used for the existing ones. See the “Zusatz zu Anlage A” for a detailed description of the new constraints and costs; the final report [1] describes how constraints are handled as part of the objective function.

1. The maximal excess of the maximal allowed number of consecutive shifts of the same shift group (among all employees)
2. The total excess of the maximal allowed number of consecutive shifts of the same shift group (sum over all employees)
3. The maximal shortfall of the minimal required number of consecutive shifts of the same shift group (among all employees)
4. The total shortfall of the minimal required number of consecutive shifts of the same shift group (sum over all employees)
5. The maximal excess of the maximal allowed number of consecutive working days (among all employees)
6. The total excess of the maximal allowed number of consecutive working days (sum over all employees)
7. The maximal shortfall of the minimal required number of consecutive working days (among all employees)
8. The total shortfall of the minimal required number of consecutive working days (sum over all employees)

3.2 Continuity at the Joints of Successive Planning Periods

To guarantee the consistency of the block structure between two successive planning periods, we consider for each employee the last block of shifts and working days of the previous planning period as part of the input for the current planning period. In particular, blocks that have been too short are tried to be extended. Nevertheless, since the algorithm cannot revise any decision made in the previous period, the shift assignments from the previous period are viewed to be fixed and any unavoidable violations of the block structure resulting from them are ignored. Likewise, we do not penalize any short-falling blocks at the end of the actual period, since they still can be extended at the beginning of the next planning period. On the other hand, we repudiate any penalizations resulting from blocks of the previous planning period that exceed the maximal allowed length. But we still try to end this block as soon as possible.

3.3 The Construction Heuristic

The initial solution of the staff scheduling problem is computed in three steps (cf. [1]): First, we compute an assignment of shifts to days and workplaces in order to meet the given minimal and maximal requirements of employees at the workplaces. Next, the employees are assigned to the shifts (and workplaces).

Finally, if breaks with flexible start times have to be considered, we plan the breaks in a greedy fashion. The so computed solution is then used as a *starting solution* for the local search procedure.

Only the second step is of relevance for the consideration of the block constraints. For each day the available employees are assigned to the shifts. This is a so called *matching problem* for each day, and it can as well be modeled and solved as a *minimum cost flow problem*. The costs of assigning an employee to a shift at a certain day are computed (heuristically) from the given input data and from the computed assignments of the previous days. These costs now also depend on the specifications of the desired block structure.

In order to enhance the chance of deriving a feasible block structure, the costs of arcs in the Min-Cost-Flow graph that are associated with an employee-to-shift assignment that would imply a violation or further violation of the new block constraints, are augmented. That is, whenever the selection of an arc would result in a block that is too long, the costs of this arc are increased by the internal costs of constraint (2) or (6), respectively. If a current block is not long enough, the selection of any other arc, corresponding to a shift of another shift group (or corresponding to the free shift if the employee is on duty the considered day and the minimal number of consecutive working days has not been reached yet) is hardened by increasing the costs of those arcs by the internal costs of constraint (4) (or (8), respectively).

3.4 Local Search

The underlying solution procedure is a modification of the simulated annealing algorithm as described in [1]. Here, the new components of the objective function require additional effort for the computation of the internal cost function as described in the following section.

3.4.1 Computation of the Internal Block Cost Function

After we have computed an initial schedule with the help of the construction heuristic, we determine its violations of constraints and calculate its objective function. Consequently the additional components of the objective function, the violations of the block structure, have to be included into this calculation as well. To determine the violations of the requirements for a block structure in the given schedule, we implemented a new schedule scanning function.

For each employee the given schedule is scanned from the beginning to the end. Whenever a change of the shift group between two consecutive days occurs, a block (of consecutive days) of shifts (of the same group) is completed and its violations, if any, of the block structure requirements can be computed.

Similarly, whenever there is a change from a working shift to the free shift, a block of consecutive working days is completed and its violation can be computed in the same way.

Note that a period of days off is viewed as a block of a special shift group and has no separate component in the objective function.

3.4.2 Ignoring Violations Forced by Fixed Shifts

Some violations of the block structure cannot be avoided since they are intrinsic by the input, e.g. by long holiday periods or other fixed shift assignments. Most of these inconsistencies are not trivial to detect. Fortunately, these violations occur rather rarely.

Two special cases of inescapable violations can be detected very efficiently. Since they concern very common “violations”, like for instance extended holidays, we decided to ignore these violations in both the computation of the internal cost function and the feasibility check. Therefore we implemented the following two ignoring rules:

Rule `NOOVER` If a block of consecutive working days/shifts of the same shift group is too long, and for each day of this block the shift assignment is fixed, then ignore the block violation.

Rule `NOUNDER` If a block contains a fixed shift assignment and it is surrounded by two other fixed shift assignments of another shift group, then ignore the violation of the minimal requirement of the enclosed block.

The rule `NOOVER` is reasonable because the algorithm cannot shorten the block since each day of the block is fixed.

On the other hand, if the days bordering a given block are fixed, the block cannot be lengthened. If additionally at least one day within the block is fixed, the block cannot be dissolved and will always cause a violation. This justifies the rule `NOUNDER`.

These two rules are also used to guarantee continuity at the beginning of the schedule (see section 3.2).

3.4.3 Recomputation of the Costs

The local search procedure tries to improve upon the given starting solution – in terms of the internal cost function value– by performing local changes to the schedule, also called *moves*. The implemented neighborhood is defined as follows: Two schedules are neighbors if they are identical except for the shift and/or workplace assignment of one employee at one day. Thus, to choose a candidate for a neighboring schedule, we switch the shift of a randomly selected employee on a randomly selected day to another shift. For the calculation of the costs of this neighboring schedule, we have to analyze the changes to the blocks before and after the selected day of this particular employee and its implications to the costs.

If an over-long block gets destroyed or a block that was too short, gets extended, which coincides with the maximal excess or maximal short-fall of block length for the selected employee, the whole schedule for this employee has to be scanned to determine his new maximal excess or maximal short-fall of block length. If necessary, the overall maximum of excess/shortfall over all employees is then updated.

Consequently, the operating expense to move from one schedule to the next cannot be done in constant time anymore. Instead the time needed to recompute the internal cost for the block structure is linear in the length of the planning period and the number of employees.

4 Performance of the Algorithm

4.1 Computational Setup

The code is implemented according to ANSI C standards. We utilized the GNU compiler gcc version 2.95.3, and we used the -O3 compiler optimization. We have tested the algorithm on a SUN Ultra Sparc II, with 448 MHz clock pulse, operating under Solaris 2.8.

4.2 Test Instances

To test the new algorithm, we modified the given instances A and B from ATOSS Software AG to enable the new criteria to take effect. The resulting instances are henceforth named A^{block} and B^{block}, respectively. The modified and new input tables are displayed in Appendix C of this document.

Andi: Also, bei Instanz B kann man das so machen. Für Instanz A gibt es ganz viele Variationen der Zielfunktion, die anderen Tabellen sind jedoch stets gleich. Hier muss man noch mal unterscheiden in A_0^{block} bis A_5^{block}.
A_0^{block}: Das ist die original Zielfunktion.
A_1^{block}: Hier sind zu den originalen Zielfunktionswerten Blockkosten, deren Gewichte halbsogross wie in

Zeile 19-26 im Anhang C sind, hinzugekommen.
A_2^{block}: Die Block-Kosten wie in Anhang C, die anderen Parameter aber wie im Original
A_3^{block}: Alles wie in Anhang C
A_4^{block}: Wie 1. oder 2.? Aber die max-dev Werte der Blockkomponenten sind durch Hohe Gewichte ersetzt.
A_5^{block}: Hier sind nur die Blockkosten wie in C, alle anderen ZF-Parameter sind auf 0 gesetzt.

Andi: Dann gibt es noch weitere Modifikationen von A, die so gemacht sind wie im alten Report nur noch die Blockkosten in der ZF wie in Anhang C haben. Und eine Grosse Instanz mit 100 Mitarbeitern.

4.3 Influence of the New Parameters on the Solution

We now compare various solutionsthat result from the two modified instances A^{block} and B^{block} from ATOSS Software AG with respect to varying values of the parameters in the objective function.

4.3.1 Influence on the Solution Quality – or: "Block Structure vs. Wishes"

Andi: Das ist zwar mieses Englisch, aber ich wollte es erst mal inhaltlich aufschreiben.

Although the algorithm hardly computes feasible solutions with respect to the block constraints if there are some maximal deviations given, the results show that the block structure appears widely in the schedule if the parameters of the blockstructure are choosen the right way.

There is a trade off between the fullfillment of the block structure and other objective function criterias especially the wishes of the employees.

First we tested the behaviour of the algorithm for different combinations of objective function parameters with instance A^{block} an compared the violations of the constraints for bothe the solution of the constructive heuristic and the best found solution of the local search algorithm within a 'time limit' of 50. The results are shown in tables 4 to 11.

Andi: Vielleicht sollte man diese Tabellen und die Bilder der Schedules einfach alle in den Anhang packen. Hier wird das echt zu viel!!! Aber man kann an den Tabellen sehen, wie sich die violations verlagern!

4.3.2 Influence on the User Given 'Time Limit'

Andi: Wie wirkt sich das auf die Laufzeit bei gegebenem Timelimit aus?
1. Mehr Computational Overhead ⇒ längere Laufzeit bei gleicher Ziefunktion (ca. 25% länger).
2. Mehr Componenten in ZF ⇒ Höherer Wert der ZF ⇒ mehr outer Iterations ⇒ längere Laufzeit.

Caro: Tabelle.

Caro: , Andi: , Berit: Berit Auswertung.

Table 1: Results for instances A^{block} and B^{block} in terms of solution quality (objective function value) and violations of wishes and the block structure for a given ‘time limit’ of 50.

instance name	Instanz-A_no_blocks	Instanz-A_blocks_reduced	Instanz-A_blocks	Instanz-A_blocks_only	Instanz-B_original	Instanz-B_blocks
feas. of start solution	🟥	🟥	🟥	🟥	🟥	🟥
feas. of local search solution	🟩	🟥	🟥	🟩	🟩	🟥
Impr. of internal costs (%)	99.7	88.8	84.1	98.2	99.8	90.2
Impr. of objective value (%)	77.3	85.0	88.2	97.2	73.7	73.6
CPU (min)	1.6	4.0	4.2	2.5	0.6	1.5
Wishes “no mod” not fullfilled	0	1	3	26	0	14
Wishes “mod” not fullfilled	0	0	0	0	0	0
shifts over max	6	2	5	0	6	5
shifts under max	3	2	2	1	3	1
shifts over toatal	29	14	13	0	64	49
shifts under total	508	24	19	10	561	10
working days over max	1	1	1	3	5	0
working days under max	4	2	2	2	4	3
working days over total	2	1	2	6	11	0
working days under total	329	10	5	2	546	66

Table 2: Solutions in dependence of the 'time limit' for Instanz A^{blocks}

	start	'time limit'										
		0	1	10	20	30	40	50	60	70	80	90
Instanz-A_no_blocks												
final solution value	4560	3185	1428	1468	1250	1142	982	1033	1031	968	856	896
final internal cost	348567	2966	1261	1311	1111	1022	866	899	897	844	744	802
final feasibility	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
CPU(sec.)		0	5	9	17	31	55	95	169	304	501	943
# outer iterations		2	7	11	20	35	59	106	191	334	572	1052
last progress in iter.		1	6	10	18	34	57	92	171	299	428	583
Instanz-A_blocks_reduced												
final solution value	63875	23929	12593	11917	12458	11656	10490	9553	9336	8040	9851	9373
final internal cost	163639	50503	25857	25227	25740	19948	18932	18360	18061	16956	18718	18076
final feasibility	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
CPU(sec.)		1	11	22	41	76	136	238	421	732	1393	2491
# outer iterations		2	11	21	38	67	121	216	387	687	1252	2231
last progress in iter.		1	7	20	23	36	78	148	247	390	273	674
Instanz-A_block												
final solution value	126040	41036	23053	22184	19391	18411	17155	14922	16468	12721	14846	13840
final internal cost	134811	51716	37235	32933	24165	20792	18840	21415	18659	16480	17504	16895
final feasibility	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
CPU(sec.)		0	11	21	42	73	135	250	440	789	1410	2531
# outer iterations		2	12	21	38	68	121	222	389	684	1238	2205
last progress in iter.		1	8	15	23	34	73	134	262	419	614	656
Instanz-A_blocks_only												
final solution value	122900	25550	6200	6200	5250	6000	6000	3500	3000	4600	4000	3850
final internal cost	99648	37424	12866	3100	12391	12766	12766	1750	1500	2300	2000	1925
final feasibility	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩
CPU(sec.)		1	10	14	37	68	130	151	258	520	935	1561
# outer iterations		2	10	14	34	62	113	131	224	452	780	1383
last progress in iter.		1	9	13	22	25	73	66	221	99	257	383

Table 3: Solutions in dependence of the ‘time limit’ for Instanz B^{block}.

	start	‘time limit’										
		0	1	10	20	30	40	50	60	70	80	90
Instanz-B_original												
final solution value	3004	5536	2255	1683	1368	1177	836	789	666	705	737	638
final internal cost	472687	7411	2443	2001	1719	1260	859	857	685	711	760	640
final feasibility	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
CPU(sec.)		0	2	4	7	13	21	37	61	111	209	347
# outer iterations		2	8	13	22	37	59	105	173	314	576	967
last progress in iter.		1	7	12	21	26	51	71	69	142	567	143
Instanz-B_blocks												
final solution value	98914	77689	47506	41905	35441	28474	27599	26073	21269	21065	21771	20284
final internal cost	191023	88204	39847	36762	33507	34933	19552	18808	16589	11261	16707	11092
final feasibility	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟩
CPU(sec.)		0	4	8	15	27	47	87	150	254	489	785
# outer iterations		2	12	22	40	73	121	217	382	639	1228	2046
last progress in iter.		1	11	17	33	43	98	103	286	298	524	824

5 Further Details of the Implementation

5.1 Scaling of Components in the Objective Function

Due to the nature of the assignments of one shift per day, each violation of the block structure is counted in units of one day. Thus the block violations can be directly compared to all other components of the objective function and further scaling is not necessary.

5.2 Objective Function Value and Internal Costs

Each component of the objective function comes either with a weight, or with an upper bound *ub* on the violation of the respective constraint. The computation of the internal weights is similar to the one used for all other components and described in detail in [1]. Generally, if there is an upper bound given for a component of the objective function, the internal weight is computed according to

$$\texttt{SCALING\_FACTOR\_WEIGHT} + \texttt{SCALING\_FACTOR\_DEV} * f\left(\frac{\max - ub}{\max}\right),$$

where *max* is the (theoretically) maximal possible violation of the constraint, and $f(\cdot)$ is a monomial.

In contrast to the computation of all other internal weights, we decided to choose $f(x) = x$ instead of $f(x) = x^{10}$. This decision is based on first computational results showing that the choice of $f(x) = x^{10}$ prevented the local search procedure to split a very long block of consecutive equivalent shifts by simply changing a single day in the middle of the block.

5.3 Documentation

Besides this document, the delivered source code is again equipped with special documentation segments in the *JavaDoc style*. Moreover, this document is accompanied by the "Zusatz zu Anlage A" for a detailed description of the input format and by the final report [1] to which this document is an extension to.

5.4 Messages

A list of the additional *error* and *warning* messages generated by the procedure is given in Appendix B.

As mentioned in Section 3.4.2, some violations of the block structure are predetermined by the input and can therefore not be avoided. Those violations that are efficiently to detect are ignored during the process of the algorithm.

References

[1] Carola Schaad, Lars Stolletz, Frederik Stork und Marc Uetz, Efficient Algorithms for Automated Staff Scheduling - Final Report, Internal Report, Technische Universität Berlin, May 2001.

Appendix

A Brief Description of new and modified Files

This is a complete list of all additional files and all files which have been modified to implement the block structure feature. For further information about the contents and the grouping of the files we refer to the final report [1].

1. Problem instance and problem instance access functions.

   (a) `staff_scheduling_problem.h`
   Has been enhanced, some data structure now contain information about shift groups (according to Table 12 of Zusatz zu Anlage A), the previos planning period (according to Table 10 of Zusatz zu Anlage A), and the minimum and maximum number of consecutive workdays.
   New access functions have been created to gather these new information.
   Includes a new function to test whether or not an employee has a fixed shift assignment for a given day.

   (j) `shift.h`
   Data structure and access functions enhanced by information about affiliation to shift group (Table 11 of Zusatz zu Anlage A, column “Schichtgruppe”).

   (k) `cost_function.h`
   The number of components of the cost function increased from 18 to 26.

   (l) `read.h`
   New keywords defined.
   `read.c`
   `read_table10` replaced.
   `read_table11` additionally reads shift group information.
   `read_table12` reads table 12 (Zusatz zu Anlage A).
   New function `read_table13` replaces old `read_table12` and reads additionally new components of the objective function.

   (m) `shift_groups.h`
   Data structure and access functions for information about shift groups according to table 12 (Zusatz zu Anlage A).

   (n) `prev_period_data.h`
   Data structure and access functions for information about the previous planning period according to Table 10 (Zusatz zu Anlage A)

2. The main functions:

   (b) `staff_schedule.h`,
   `staff_schedule.c`
   Two new functions for schedule visualization.

   (c) `solve_staff_scheduling_problem.c`
   Calls `init_internal_weights` before start heuristic.
   Extended output for schedule visualization.

3. The constructive heuristic:

(c) `assign_empl.h`,
`assign_empl.c`
Parameter and implementation of various functions changed to consider block structure (see section 3.3).

4. Local search and cost:

   (a) `local_search.c`
   Determination and evaluation of the blocks of a schedule before and after a given day for a given employee, see section 3.4.3.

   (b) `objective_function.h`
   New components and access functions for the violations of the block structure in the data structures for internal weights and violations.
   New parameter for the function `recompute_internal_cost`. `objective_function.c`
   Various new and extended functions to compute and recompute violations and costs of the block structure of a schedule.
   Various extended access functions for violations and internal weights.
   Extended output of the objective function.

   (c) `curr_shift_data.h`
   New data structure and access functions for information about the block structure of a schedule before and after a given day for a given employee.

5. Plausibility and messages

   (a) `plausibility.c`
   Extension of the plausibility checks.

   (b) `error_message.h`,
   `error_message.c`
   There are several new errors and warnings originated by the new components for block structure. A new assignment of error and warning numbers was necessary. The meaning of all error IDs is documented in Appendix B.

6. Miscellaneous

   No changes are made in the files of this group.

B List of new the error messages

In the following we describe the new messages in detail.

47. ERROR_PREVPERIOD_SHIFT
    a shift index of the last day of the previous period is out of range
    Additional info: (wrong) shift index, tablenumber as in extension of Anlage A, employee index

48. ERROR_PREVPERIOD_NUMB_SHIFT
    the number of consecutive ocurrences of the last shift of the previous period is negative
    Additional info: (wrong) number of the shift, tablenumber as in extension of Anlage A, employee index

49. ERROR_PREVPERIOD_NUMB_WORKDAYS
the number of consecutive working shifts until the last day of the previous period is negative
Additional info: (wrong) number of the working days, tablenumber as in extension of Anlage A, employee index

50. ERROR_SHIFT_GROUP
a group index of a shift is out of range
Additional info: (wrong) shift group index, tablenumber as in extension of Anlage A, shift index

51. ERROR_SHIFTGROUP_GROUPINDEX
a shift group index is out of range
Additional info: (wrong) shift group index, tablenumber as in extension of Anlage A

52. ERROR_SHIFTGROUP_MIN
the minimal required number of consecutive shifts of a group is negative
Additional info: (wrong) minimal required number of shifts, tablenumber as in extension of Anlage A, shift group index

53. ERROR_SHIFTGROUP_MAX
the maximal allowed number of consecutive shifts of a group is negative
Additional info: (wrong) maximal allowed number of shifts, tablenumber as in extension of Anlage A, shift group index

54. ERROR_SHIFTGROUP_MINMAX
the minimal required number of shifts is greater than the maximal allowed number of shifts of a group
Additional info: minimal required number of shifts, maximal allowed number of shifts, tablenumber as in extension of Anlage A, shift group index

-1.- -24. WARNING_OVERPLAN_EMPL,
WARNING_UNDERPLAN_EMPL,
WARNING_OVERPLAN_TOTAL,
WARNING_UNDERPLAN_TOTAL,
WARNING_OVERPLAN_WP,
WARNING_UNDERPLAN_WP,
WARNING_NOWISH_MOD,
WARNING_NOWISH_NOMOD,
WARNING_REST_TIME,
WARNING_QUALIFICATION,
WARNING_VOL_OVER,
WARNING_VOL_UNDER,
WARNING_BALANCE_OVER_EMPL,
WARNING_BALANCE_UNDER_EMPL,
WARNING_BALANCE_OVER_TOTAL,
WARNING_BALANCE_UNDER_TOTAL,
WARNING_CONS_SHIFTS_OVER_MAX,
WARNING_CONS_SHIFTS_OVER_TOTAL,
WARNING_CONS_SHIFTS_UNDER_MAX,
WARNING_CONS_SHIFTS_UNDER_TOTAL,
WARNING_CONS_WORKDAYS_OVER_MAX,
WARNING_CONS_WORKDAYS_OVER_TOTAL,

WARNING_CONS_WORKDAYS_UNDER_MAX,
WARNING_CONS_WORKDAYS_UNDER_TOTAL

C Test Instances

C.1 Instance A^{block}

Tabelle 10: Arbeitsende im vorherigen Planungszeitraum

Mitarbeiterindex *	Arbeitsmuster *	Anzahl_Arbeitsmuster *	Anzahl_Arbeitstage
1	4	1	0
2	4	1	0
3	9	2	3
4	2	1	1
5	3	1	4
6	2	1	2
7	1	1	1
8	1	1	1
9	10	1	4
10	3	1	6
11	3	1	1
12	4	3	0
13	4	5	0
14	11	1	1
15	4	1	0
16	1	1	1
17	2	1	1
18	4	7	0

```
Tabellenende


Tabelle 11: Arbeitsmuster
Arbeitsmusterindex * Gruppe  * von   *    bis  *    Sollzeit  *  Pausen

         1           1         360         840          450     0 0 0  0 0 0  0 0
         2           2         840        1320          450     0 0 0  0 0 0  0 0
         3           3        1320        1800          450     0 0 0  0 0 0  0 0
         4           4           0           0            0     0 0 0  0 0 0  0 0
         5           5         540         930          360     0 0 0  0 0 0  0 0
         6           6         480        1020          480     0 0 0  0 0 0  0 0
         7           7         480         780          300     0 0 0  0 0 0  0 0
         8           8         780        1080          300     0 0 0  0 0 0  0 0
         9           9         240         720          450     0 0 0  0 0 0  0 0
        10          10         720        1200          450     0 0 0  0 0 0  0 0
        11          11        1200        1680          450     0 0 0  0 0 0  0 0
Tabellenende

Tabelle 12: Arbeitsmuster-Gruppen

Arbeitsmuster-Gruppen-Index * Min * Max

         1           2            7
         2           4            6
         3           4            6
         4           1            3
         5           2            7
         6           2            7
         7           4            7
         8           4            7
         9           4            8
        10           4            6
        11           3            6
Tabellenende



Tabelle 13

 Typ der Funktion     *       Gewichtung    *    Maximale Abweichung vom Sollwert

       1                         50                          -
       2                         60                          -
       3                         120                         -
       4                         0                           -
       5                         10                          -
       6                         70                          -
```

```
    7                    20                         -
    8                    100                        -
    9                    0                          0
    10                   80                         -
    11                   0                          -
    12                   0                          -
    13                   0                          -
    14                   0                          -
    15                   0                          -
    16                   0                          -
    17                   0                          -
    18                   0                          -
    19                   0                          5
    20                   0                          1
    21                   100                        -
    22                   200                        -
    23                   0                          4
    24                   0                          2
    25                   200                        -
    26                   150                        -
Tabellenende
```

C.2 Instance B^{block}

```
Tabelle 10: Arbeitsende im vorherigen Planungszeitraum


Mitarbeiterindex  *   Arbeitsmuster * Anzahl_Arbeitsmuster * Anzahl_Arbeitstage

          1           3           1           2

          2           2           1           0

          3           4           1           2

          4           2           2           0

          5           1           1           1

          6           1           1           1

          7           3           1           1

          8           1           1           1

          9           3           1           1
```

```
        10          2          1          0

        11          2          2          0

        12          2          3          0

        13          3          1          1

        14          1          1          1

        15          4          3          3

        16          2          2          0

        17          1          2          2

        18          2          1          0

        19          2          1          0

        20          1          3          3

        21          2          2          0

        22          4          2          2

        23          3          1          3

        24          2          1          0

        25          1          1          1

        26          4          1          1

        27          3          1          1

        28          2          1          0

Tabellenende
Tabelle 11: Arbeitsmuster
Arbeitsmusterindex * Gruppe  * von   *    bis  *    Sollzeit  *  Pausen

         1           1          360          840          450    0 0 0  0 0 0  0 0
         2           2            0            0            0    0 0 0  0 0 0  0 0
         3           3         1320         1800          450    0 0 0  0 0 0  0 0
         4           4          840         1320          450    0 0 0  0 0 0  0 0
         5           5          540          930          360    0 0 0  0 0 0  0 0
Tabellenende
```

Tabelle 12: Arbeitsmuster-Gruppen

Arbeitsmuster-Gruppen-Index	Min	Max
1	2	7
2	2	3
3	4	6
4	3	6
5	2	7

Tabellenende

Tabelle 13

Typ der Funktion	Gewichtung	Maximale Abweichung vom Sollwert
1	50	-
2	60	-
3	0	3000
4	0	-
5	10	-
6	70	-
7	20	-
8	100	-
9	0	0
10	80	-
11	0	-
12	30	-
13	0	-
14	30	-
15	0	-
16	0	2400
17	0	-
18	0	-
19	0	5
20	0	1
21	100	-
22	200	-
23	0	4
24	0	2
25	200	-
26	150	-

Tabellenende

D Computational Results

D.1 Distribution of costs & solution schedules

In these tables, the first column displays the criterion of the objective function (see Table 13 of Zusatz zu Anlage A), the second is the violation of the criterion in days, the third is the same violation in minutes, then the weight or maximal deviation, and finally the cost.

To represent the block structure of the solutions for instance A^{block}, we picture the referring schedules in figures 1 to 8.

D.1.1 Instance A^{blocks}

Table 4: Distribution of the costs in the start solution, Instance A, original

	criterion	viol (days)	viol (min)	weight/dev		cost
1	mx overplan empl:	0.94 units	(450 min)	w	50	46.88
2	mx underplan empl:	6.56 units	(3150 min)	w	60	393.75
3	overplan total:	0.00 units	(0 min)	w	120	0.00
4	underplan total:	22.21 units	(10660 min)	w	0	0.00
5	overplan workplace:	0.00 units	(0 min)	w	10	0.00
6	underplan workplace:	0.00 units	(0 min)	w	70	0.00
7	wish type 0:	16.00 units	(16 day)	w	20	320.00
8	wish type 1:	22.00 units	(22 day)	w	100	2200.00
9	rest time:	33.75 units	(16200 min)	d	0	0.00
10	qualification:	20.00 units	(9600 min)	w	80	1600.00
12	volume overplan:	0.00 units	(0 min)	w	0	0.00
14	volume underplan:	0.00 units	(0 min)	w	0	0.00
17	bal overplan total:	9.85 units	(4730 min)	w	0	0.00
18	bal underplan total:	0.00 units	(0 min)	w	0	0.00
19	mx shifts over:	21.00 units	(21 day)	w	0	0.00
20	mx shifts under:	3.00 units	(3 day)	w	0	0.00
21	shifts over total:	49.00 units	(49 day)	w	0	0.00
22	shifts under total:	539.00 units	(539 day)	w	0	0.00
23	mx workays over:	8.00 units	(8 day)	w	0	0.00
24	mx workdays under:	4.00 units	(4 day)	w	0	0.00
25	workdays over total:	21.00 units	(21 day)	w	0	0.00
26	workdays under total:	358.00 units	(358 day)	w	0	0.00
	total costs					4560.62

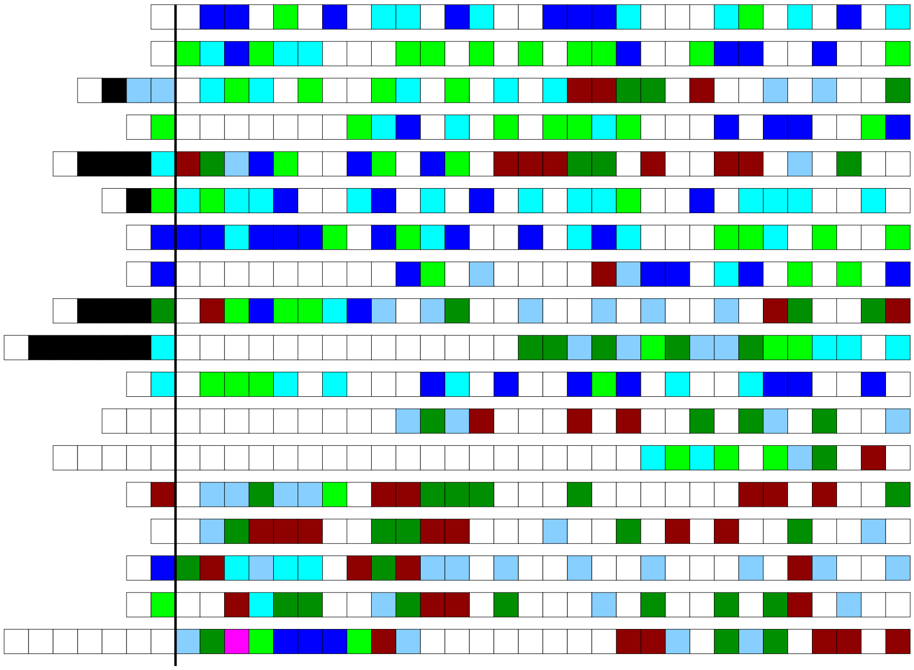

Figure 1: Start solution of Instance A, original.

Table 5: Distribution of the costs in the final solution, Instance A, original

	criterion	viol (days)	viol (min)	weight/dev		cost
1	mx overplan empl:	1.88 units	(900 min)	w	50	93.75
2	mx underplan empl:	6.40 units	(3070 min)	w	60	383.75
3	overplan total:	0.00 units	(0 min)	w	120	0.00
4	underplan total:	1.90 units	(910 min)	w	0	0.00
5	overplan workplace:	1.81 units	(870 min)	w	10	18.12
6	underplan workplace:	2.25 units	(1080 min)	w	70	157.50
7	wish type 0:	7.00 units	(7 day)	w	20	140.00
8	wish type 1:	0.00 units	(0 day)	w	100	0.00
9	rest time:	0.00 units	(0 min)	d	0	0.00
10	qualification:	3.00 units	(1440 min)	w	80	240.00
12	volume overplan:	0.00 units	(0 min)	w	0	0.00
14	volume underplan:	0.00 units	(0 min)	w	0	0.00
17	bal overplan total:	30.17 units	(14480 min)	w	0	0.00
18	bal underplan total:	0.00 units	(0 min)	w	0	0.00
19	mx shifts over:	6.00 units	(6 day)	w	0	0.00
20	mx shifts under:	3.00 units	(3 day)	w	0	0.00
21	shifts over total:	29.00 units	(29 day)	w	0	0.00
22	shifts under total:	508.00 units	(508 day)	w	0	0.00
23	mx workays over:	1.00 units	(1 day)	w	0	0.00
24	mx workdays under:	4.00 units	(4 day)	w	0	0.00
25	workdays over total:	2.00 units	(2 day)	w	0	0.00
26	workdays under total:	329.00 units	(329 day)	w	0	0.00
	total costs					1033.12

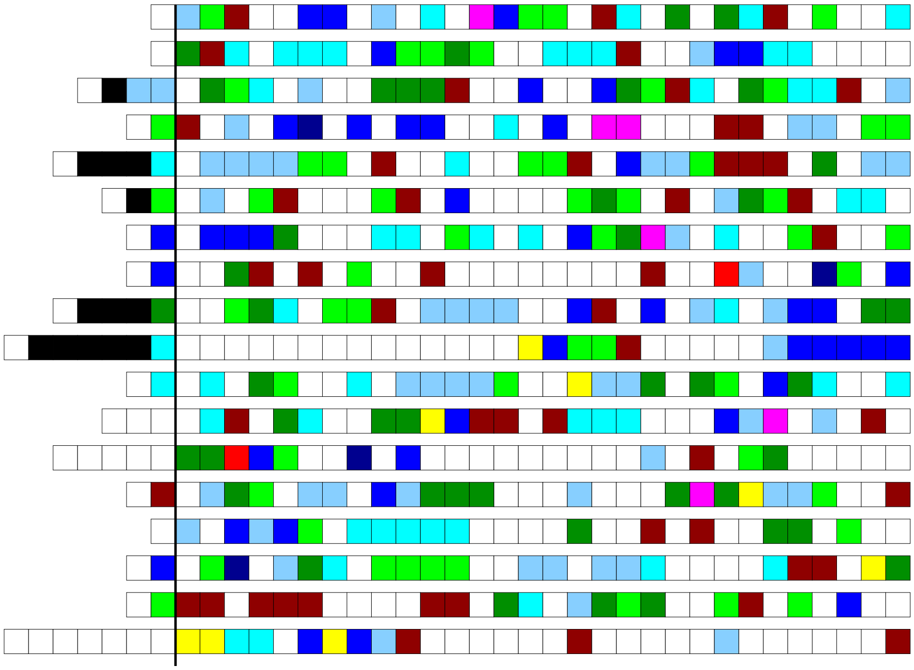

Figure 2: Final solution of Instance A, original.

Table 6: Distribution of the costs in the start solution, Instance A, low block weights

	criterion	viol (days)	viol (min)	weight/dev		cost
1	mx overplan empl:	0.94 units	(450 min)	w	50	46.88
2	mx underplan empl:	6.56 units	(3150 min)	w	60	393.75
3	overplan total:	0.00 units	(0 min)	w	120	0.00
4	underplan total:	22.21 units	(10660 min)	w	0	0.00
5	overplan workplace:	0.00 units	(0 min)	w	10	0.00
6	underplan workplace:	0.00 units	(0 min)	w	70	0.00
7	wish type 0:	13.00 units	(13 day)	w	20	260.00
8	wish type 1:	26.00 units	(26 day)	w	100	2600.00
9	rest time:	5.00 units	(2400 min)	d	0	0.00
10	qualification:	20.00 units	(9600 min)	w	80	1600.00
12	volume overplan:	0.00 units	(0 min)	w	0	0.00
14	volume underplan:	0.00 units	(0 min)	w	0	0.00
17	bal overplan total:	9.85 units	(4730 min)	w	0	0.00
18	bal underplan total:	0.00 units	(0 min)	w	0	0.00
19	mx shifts over:	8.00 units	(8 day)	d	5	0.00
20	mx shifts under:	3.00 units	(3 day)	d	1	0.00
21	shifts over total:	35.00 units	(35 day)	w	50	1750.00
22	shifts under total:	269.00 units	(269 day)	w	100	26900.00
23	mx workays over:	3.00 units	(3 day)	d	4	0.00
24	mx workdays under:	4.00 units	(4 day)	d	2	0.00
25	workdays over total:	4.00 units	(4 day)	w	100	400.00
26	workdays under total:	399.00 units	(399 day)	w	75	29925.00
	total costs					63875.62

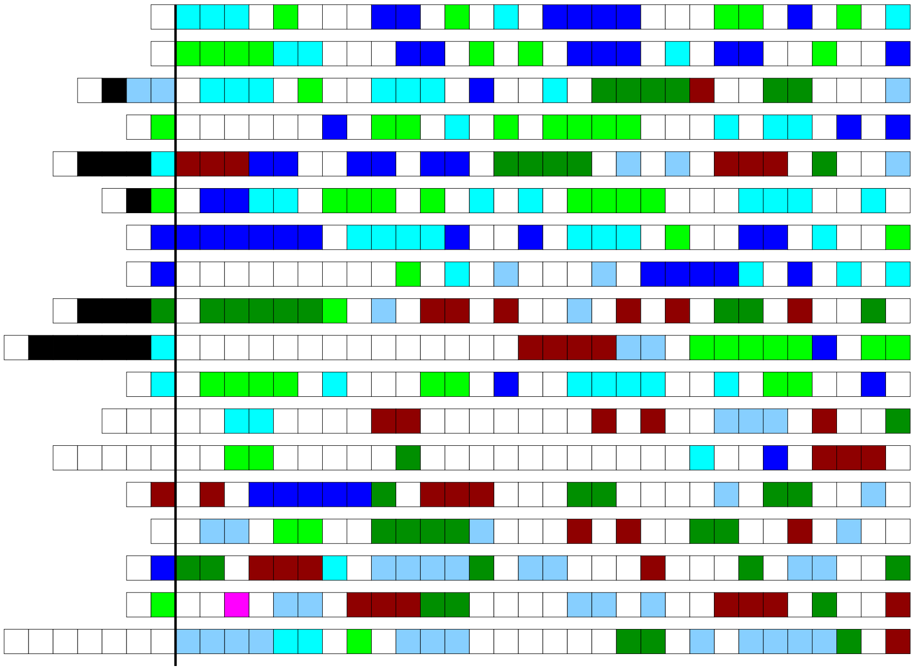

Figure 3: Start solution of instance A, low block weights.

Table 7: Distribution of the costs in the final solution, Instance A, low block weights

	criterion	viol (days)	viol (min)	weight/dev		cost
1	mx overplan empl:	2.50 units	(1200 min)	w	50	125.00
2	mx underplan empl:	6.08 units	(2920 min)	w	60	365.00
3	overplan total:	0.10 units	(50 min)	w	120	12.50
4	underplan total:	0.00 units	(0 min)	w	0	0.00
5	overplan workplace:	30.88 units	(14820 min)	w	10	308.75
6	underplan workplace:	41.75 units	(20040 min)	w	70	2922.50
7	wish type 0:	23.00 units	(23 day)	w	20	460.00
8	wish type 1:	1.00 units	(1 day)	w	100	100.00
9	rest time:	0.00 units	(0 min)	d	0	0.00
10	qualification:	16.38 units	(7860 min)	w	80	1310.00
12	volume overplan:	0.00 units	(0 min)	w	0	0.00
14	volume underplan:	0.00 units	(0 min)	w	0	0.00
17	bal overplan total:	32.17 units	(15440 min)	w	0	0.00
18	bal underplan total:	0.00 units	(0 min)	w	0	0.00
19	mx shifts over:	2.00 units	(2 day)	d	5	0.00
20	mx shifts under:	2.00 units	(2 day)	d	1	0.00
21	shifts over total:	14.00 units	(14 day)	w	50	700.00
22	shifts under total:	24.00 units	(24 day)	w	100	2400.00
23	mx workays over:	1.00 units	(1 day)	d	4	0.00
24	mx workdays under:	2.00 units	(2 day)	d	2	0.00
25	workdays over total:	1.00 units	(1 day)	w	100	100.00
26	workdays under total:	10.00 units	(10 day)	w	75	750.00
	total costs					9553.75

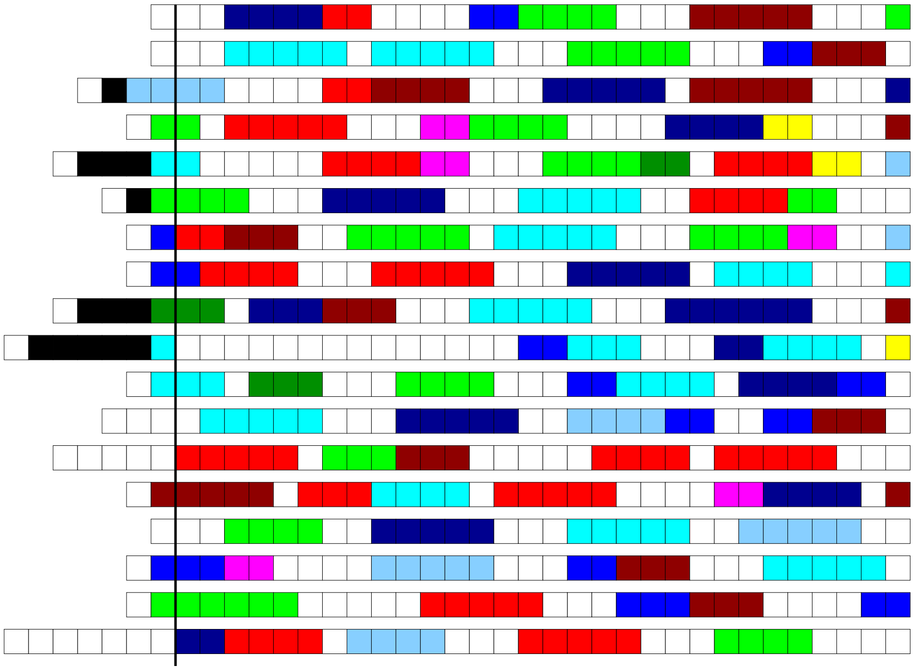

Figure 4: Final solution of instance A, low block weights.

Table 8: Distribution of the costs in the start solution, Instance A, high block weights

	criterion	viol (days)	viol (min)	weight/dev		cost
1	mx overplan empl:	0.94 units	(450 min)	w	50	46.88
2	mx underplan empl:	6.56 units	(3150 min)	w	60	393.75
3	overplan total:	0.00 units	(0 min)	w	120	0.00
4	underplan total:	22.02 units	(10570 min)	w	0	0.00
5	overplan workplace:	0.00 units	(0 min)	w	10	0.00
6	underplan workplace:	0.00 units	(0 min)	w	70	0.00
7	wish type 0:	20.00 units	(20 day)	w	20	400.00
8	wish type 1:	22.00 units	(22 day)	w	100	2200.00
9	rest time:	2.25 units	(1080 min)	d	0	0.00
10	qualification:	20.00 units	(9600 min)	w	80	1600.00
12	volume overplan:	0.00 units	(0 min)	w	0	0.00
14	volume underplan:	0.00 units	(0 min)	w	0	0.00
17	bal overplan total:	10.04 units	(4820 min)	w	0	0.00
18	bal underplan total:	0.00 units	(0 min)	w	0	0.00
19	mx shifts over:	7.00 units	(7 day)	d	5	0.00
20	mx shifts under:	3.00 units	(3 day)	d	1	0.00
21	shifts over total:	38.00 units	(38 day)	w	100	3800.00
22	shifts under total:	269.00 units	(269 day)	w	200	53800.00
23	mx workays over:	3.00 units	(3 day)	d	4	0.00
24	mx workdays under:	4.00 units	(4 day)	d	2	0.00
25	workdays over total:	4.00 units	(4 day)	w	200	800.00
26	workdays under total:	420.00 units	(420 day)	w	150	63000.00
	total costs					126040.62

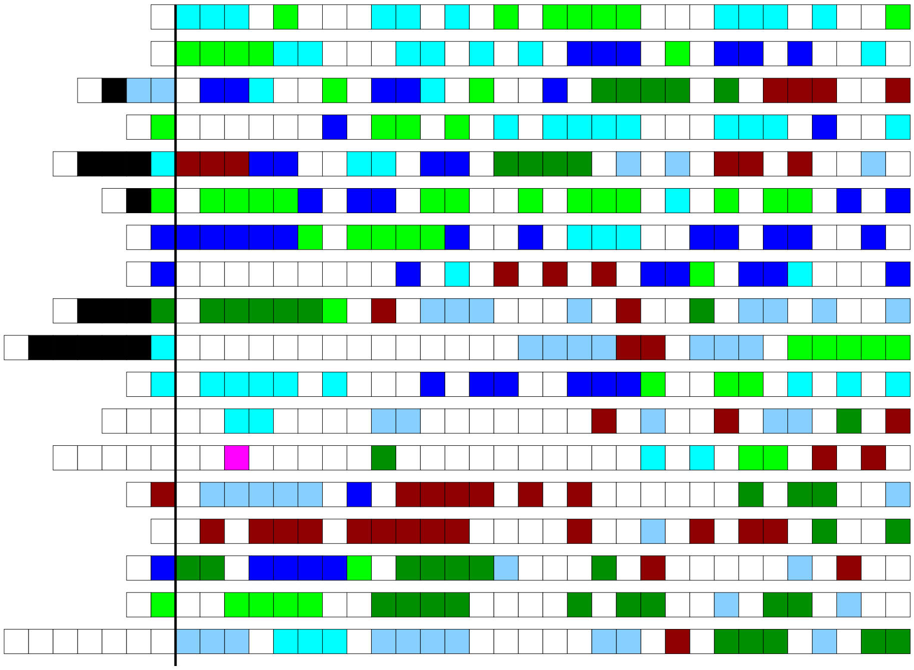

Figure 5: Start solution of instance A, high block weights.

Table 9: Distribution of the costs in the final solution, Instance A, high block weights

	criterion	viol (days)	viol (min)	weight/dev		cost
1	mx overplan empl:	4.76 units	(2286 min)	w	50	238.12
2	mx underplan empl:	3.56 units	(1710 min)	w	60	213.75
3	overplan total:	0.00 units	(0 min)	d	3000	0.00
4	underplan total:	29.10 units	(13968 min)	w	0	0.00
5	overplan workplace:	45.43 units	(21805 min)	w	10	454.27
6	underplan workplace:	96.39 units	(46268 min)	w	70	6747.42
7	wish type 0:	11.00 units	(11 day)	w	20	220.00
8	wish type 1:	14.00 units	(14 day)	w	100	1400.00
9	rest time:	0.00 units	(0 min)	d	0	0.00
10	qualification:	0.00 units	(0 min)	w	80	0.00
12	volume overplan:	0.00 units	(0 min)	w	30	0.00
14	volume underplan:	0.00 units	(0 min)	w	30	0.00
17	bal overplan total:	4.82 units	(2312 min)	d	2400	0.00
18	bal underplan total:	0.00 units	(0 min)	w	0	0.00
19	mx shifts over:	5.00 units	(5 day)	d	5	0.00
20	mx shifts under:	1.00 units	(1 day)	d	1	0.00
21	shifts over total:	49.00 units	(49 day)	w	100	4900.00
22	shifts under total:	10.00 units	(10 day)	w	200	2000.00
23	mx workays over:	0.00 units	(0 day)	d	4	0.00
24	mx workdays under:	3.00 units	(3 day)	d	2	0.00
25	workdays over total:	0.00 units	(0 day)	w	200	0.00
26	workdays under total:	66.00 units	(66 day)	w	150	9900.00
	total costs					26073.56

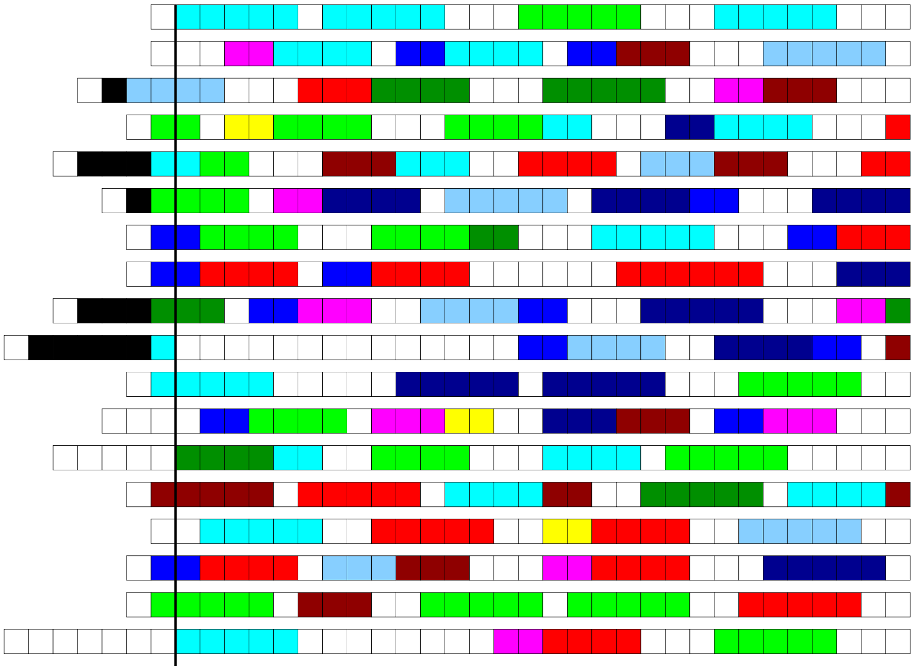

Figure 6: Final solution of instance A, high block weights.

Table 10: Distribution of the costs in the start solution, Instance A, exclusively high block weights

	criterion	viol (days)	viol (min)	weight/dev		cost
1	mx overplan empl:	0.94 units	(450 min)	w	0	0.00
2	mx underplan empl:	6.56 units	(3150 min)	w	0	0.00
3	overplan total:	0.00 units	(0 min)	w	0	0.00
4	underplan total:	22.02 units	(10570 min)	w	0	0.00
5	overplan workplace:	0.00 units	(0 min)	w	0	0.00
6	underplan workplace:	0.00 units	(0 min)	w	0	0.00
7	wish type 0:	21.00 units	(21 day)	w	0	0.00
8	wish type 1:	24.00 units	(24 day)	w	0	0.00
9	rest time:	3.25 units	(1560 min)	w	0	0.00
10	qualification:	24.00 units	(11520 min)	w	0	0.00
12	volume overplan:	0.00 units	(0 min)	w	0	0.00
14	volume underplan:	0.00 units	(0 min)	w	0	0.00
17	bal overplan total:	10.04 units	(4820 min)	w	0	0.00
18	bal underplan total:	0.00 units	(0 min)	w	0	0.00
19	mx shifts over:	6.00 units	(6 day)	d	5	0.00
20	mx shifts under:	3.00 units	(3 day)	d	1	0.00
21	shifts over total:	36.00 units	(36 day)	w	100	3600.00
22	shifts under total:	267.00 units	(267 day)	w	200	53400.00
23	mx workays over:	3.00 units	(3 day)	d	4	0.00
24	mx workdays under:	4.00 units	(4 day)	d	2	0.00
25	workdays over total:	4.00 units	(4 day)	w	200	800.00
26	workdays under total:	434.00 units	(434 day)	w	150	65100.00
	total costs					122900.00

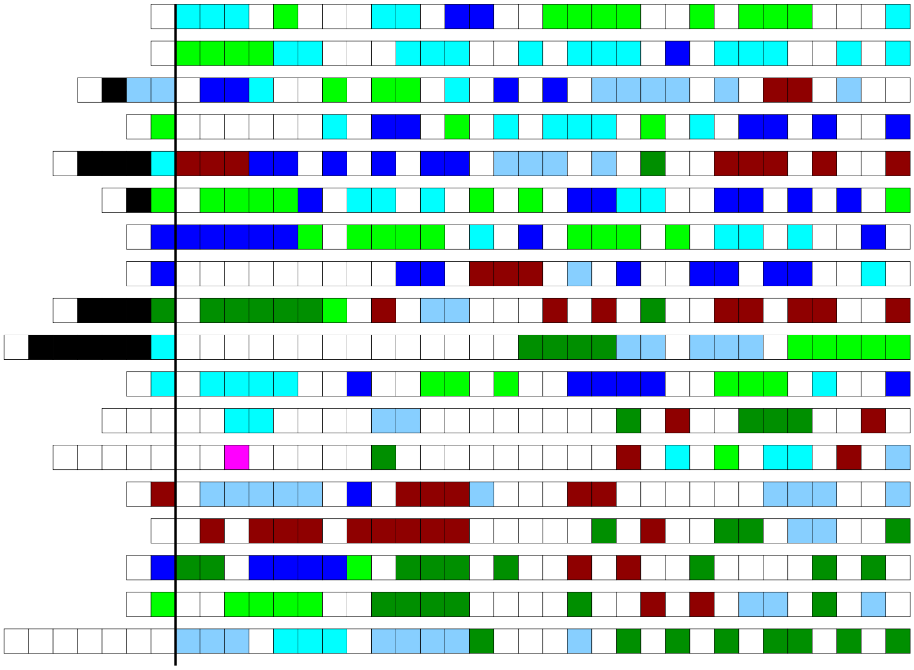

Figure 7: Start solution of instance A, exclusively high block weights.

Table 11: Distribution of the costs in the final solution, Instance A

	criterion	viol (days)	viol (min)	weight/dev		cost
1	mx overplan empl:	11.31 units	(5430 min)	w	0	0.00
2	mx underplan empl:	6.02 units	(2890 min)	w	0	0.00
3	overplan total:	78.48 units	(37670 min)	w	0	0.00
4	underplan total:	0.00 units	(0 min)	w	0	0.00
5	overplan workplace:	113.00 units	(54240 min)	w	0	0.00
6	underplan workplace:	49.81 units	(23910 min)	w	0	0.00
7	wish type 0:	30.00 units	(30 day)	w	0	0.00
8	wish type 1:	26.00 units	(26 day)	w	0	0.00
9	rest time:	10.88 units	(5220 min)	w	0	0.00
10	qualification:	32.06 units	(15390 min)	w	0	0.00
12	volume overplan:	0.00 units	(0 min)	w	0	0.00
14	volume underplan:	0.00 units	(0 min)	w	0	0.00
17	bal overplan total:	110.54 units	(53060 min)	w	0	0.00
18	bal underplan total:	0.00 units	(0 min)	w	0	0.00
19	mx shifts over:	0.00 units	(0 day)	d	5	0.00
20	mx shifts under:	1.00 units	(1 day)	d	1	0.00
21	shifts over total:	0.00 units	(0 day)	w	100	0.00
22	shifts under total:	10.00 units	(10 day)	w	200	2000.00
23	mx workays over:	3.00 units	(3 day)	d	4	0.00
24	mx workdays under:	2.00 units	(2 day)	d	2	0.00
25	workdays over total:	6.00 units	(6 day)	w	200	1200.00
26	workdays under total:	2.00 units	(2 day)	w	150	300.00
	total costs					3500.00

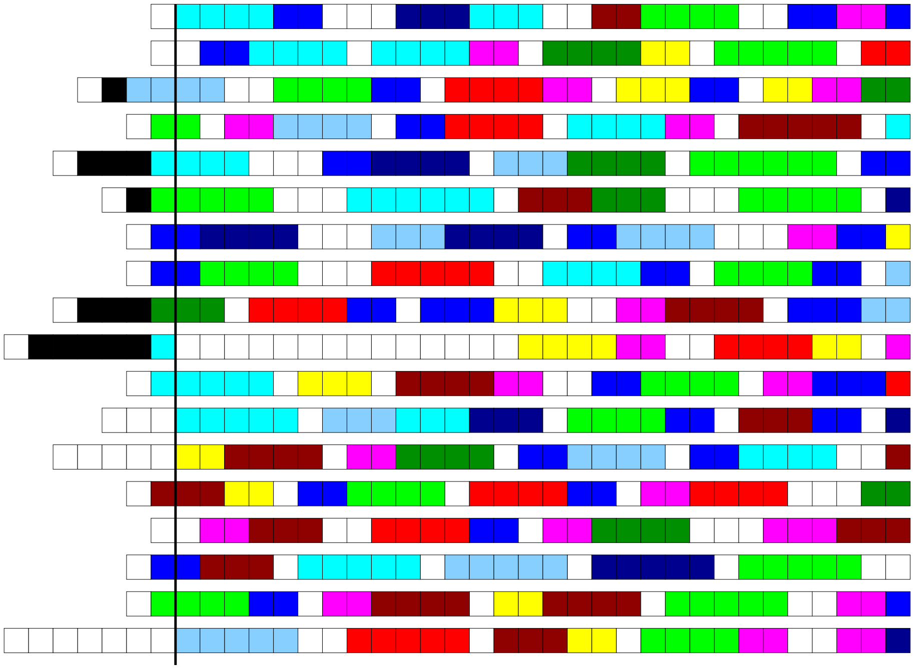

Figure 8: final solution of instance A, exclusively high block weights.

D.1.2 Instance B^{blocks}

Table 12: Distribution of the costs in the start solution, Instance B, original

	criterion	viol (days)	viol (min)	weight/dev		cost
1	mx overplan empl:	2.81 units	(1350 min)	w	50	140.62
2	mx underplan empl:	6.56 units	(3150 min)	w	60	393.75
3	overplan total:	0.00 units	(0 min)	d	3000	0.00
4	underplan total:	23.51 units	(11286 min)	w	0	0.00
5	overplan workplace:	1.00 units	(481 min)	w	10	10.02
6	underplan workplace:	0.00 units	(0 min)	w	70	0.00
7	wish type 0:	13.00 units	(13 day)	w	20	260.00
8	wish type 1:	22.00 units	(22 day)	w	100	2200.00
9	rest time:	40.88 units	(19620 min)	d	0	0.00
10	qualification:	0.00 units	(0 min)	w	80	0.00
12	volume overplan:	0.00 units	(0 min)	w	30	0.00
14	volume underplan:	0.00 units	(0 min)	w	30	0.00
17	bal overplan total:	10.40 units	(4994 min)	d	2400	0.00
18	bal underplan total:	0.00 units	(0 min)	w	0	0.00
19	mx shifts over:	6.00 units	(6 day)	w	0	0.00
20	mx shifts under:	3.00 units	(3 day)	w	0	0.00
21	shifts over total:	36.00 units	(36 day)	w	0	0.00
22	shifts under total:	833.00 units	(833 day)	w	0	0.00
23	mx workays over:	7.00 units	(7 day)	w	0	0.00
24	mx workdays under:	4.00 units	(4 day)	w	0	0.00
25	workdays over total:	18.00 units	(18 day)	w	0	0.00
26	workdays under total:	789.00 units	(789 day)	w	0	0.00
	total costs					3004.40

Table 13: Distribution of the costs in the final solution, Instance B, original

	criterion	viol (days)	viol (min)	weight/dev		cost
1	mx overplan empl:	3.26 units	(1566 min)	w	50	163.12
2	mx underplan empl:	7.88 units	(3780 min)	w	60	472.50
3	overplan total:	0.00 units	(0 min)	d	3000	0.00
4	underplan total:	29.10 units	(13968 min)	w	0	0.00
5	overplan workplace:	0.00 units	(1 min)	w	10	0.02
6	underplan workplace:	1.05 units	(505 min)	w	70	73.65
7	wish type 0:	4.00 units	(4 day)	w	20	80.00
8	wish type 1:	0.00 units	(0 day)	w	100	0.00
9	rest time:	0.00 units	(0 min)	d	0	0.00
10	qualification:	0.00 units	(0 min)	w	80	0.00
12	volume overplan:	0.00 units	(0 min)	w	30	0.00
14	volume underplan:	0.00 units	(0 min)	w	30	0.00
17	bal overplan total:	4.82 units	(2312 min)	d	2400	0.00
18	bal underplan total:	0.00 units	(0 min)	w	0	0.00
19	mx shifts over:	6.00 units	(6 day)	w	0	0.00
20	mx shifts under:	3.00 units	(3 day)	w	0	0.00
21	shifts over total:	64.00 units	(64 day)	w	0	0.00
22	shifts under total:	561.00 units	(561 day)	w	0	0.00
23	mx workays over:	5.00 units	(5 day)	w	0	0.00
24	mx workdays under:	4.00 units	(4 day)	w	0	0.00
25	workdays over total:	11.00 units	(11 day)	w	0	0.00
26	workdays under total:	546.00 units	(546 day)	w	0	0.00
	total costs					789.29

Table 14: Distribution of the costs in the start solution, Instance B, high blocks weight

	criterion	viol (days)	viol (min)	weight/dev		cost
1	mx overplan empl:	2.81 units	(1350 min)	w	50	140.62
2	mx underplan empl:	6.56 units	(3150 min)	w	60	393.75
3	overplan total:	0.00 units	(0 min)	d	3000	0.00
4	underplan total:	25.24 units	(12114 min)	w	0	0.00
5	overplan workplace:	1.00 units	(481 min)	w	10	10.02
6	underplan workplace:	0.00 units	(0 min)	w	70	0.00
7	wish type 0:	16.00 units	(16 day)	w	20	320.00
8	wish type 1:	26.00 units	(26 day)	w	100	2600.00
9	rest time:	6.38 units	(3060 min)	d	0	0.00
10	qualification:	0.00 units	(0 min)	w	80	0.00
12	volume overplan:	0.00 units	(0 min)	w	30	0.00
14	volume underplan:	0.00 units	(0 min)	w	30	0.00
17	bal overplan total:	8.68 units	(4166 min)	d	2400	0.00
18	bal underplan total:	0.00 units	(0 min)	w	0	0.00
19	mx shifts over:	6.00 units	(6 day)	d	5	0.00
20	mx shifts under:	3.00 units	(3 day)	d	1	0.00
21	shifts over total:	34.00 units	(34 day)	w	100	3400.00
22	shifts under total:	194.00 units	(194 day)	w	200	38800.00
23	mx workays over:	3.00 units	(3 day)	d	4	0.00
24	mx workdays under:	4.00 units	(4 day)	d	2	0.00
25	workdays over total:	9.00 units	(9 day)	w	200	1800.00
26	workdays under total:	343.00 units	(343 day)	w	150	51450.00
	total costs					98914.40

Table 15: Distribution of the costs in the final solution, Instance B, high block weights.

	criterion	viol (days)	viol (min)	weight/dev		cost
1	mx overplan empl:	4.76 units	(2286 min)	w	50	238.12
2	mx underplan empl:	3.56 units	(1710 min)	w	60	213.75
3	overplan total:	0.00 units	(0 min)	d	3000	0.00
4	underplan total:	29.10 units	(13968 min)	w	0	0.00
5	overplan workplace:	45.43 units	(21805 min)	w	10	454.27
6	underplan workplace:	96.39 units	(46268 min)	w	70	6747.42
7	wish type 0:	11.00 units	(11 day)	w	20	220.00
8	wish type 1:	14.00 units	(14 day)	w	100	1400.00
9	rest time:	0.00 units	(0 min)	d	0	0.00
10	qualification:	0.00 units	(0 min)	w	80	0.00
12	volume overplan:	0.00 units	(0 min)	w	30	0.00
14	volume underplan:	0.00 units	(0 min)	w	30	0.00
17	bal overplan total:	4.82 units	(2312 min)	d	2400	0.00
18	bal underplan total:	0.00 units	(0 min)	w	0	0.00
19	mx shifts over:	5.00 units	(5 day)	d	5	0.00
20	mx shifts under:	1.00 units	(1 day)	d	1	0.00
21	shifts over total:	49.00 units	(49 day)	w	100	4900.00
22	shifts under total:	10.00 units	(10 day)	w	200	2000.00
23	mx workays over:	0.00 units	(0 day)	d	4	0.00
24	mx workdays under:	3.00 units	(3 day)	d	2	0.00
25	workdays over total:	0.00 units	(0 day)	w	200	0.00
26	workdays under total:	66.00 units	(66 day)	w	150	9900.00
	total costs					26073.56