Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Computer Science

DIPLOMA THESIS AGREEMENT

Student: Král David

Study programme: Open Informatics
Specialisation: Software Engineering

Title of Diploma Thesis: Scheduling of energy-demanding operations with varying cost of electricity

Guidelines:

The goal of this thesis is to design and implement an algorithm for scheduling energy-demanding operations in production w.r.t. the varying cost of the electricity. In contrast with the existing research [3], the aim of this thesis is to propose an efficient exact algorithm. The following tasks have to be done: Study the problem of energy-aware scheduling and review the existing works.Choose an appropriate approach for solving the given scheduling problem.Design and implement an algorithm for the given scheduling problem.Test the implemented algorithm on randomly generated instances and compare it with a Mixed Integer Linear Programming model. The instances should cover a broad range of possible situations, e.g. equal/arbitrary processing times, arbitrary release times/zero release times etc.

Bibliography/Sources:

[1] Maximilian Selmair, Thorsten Claus, Marco Trost, Andreas Bley, Frank Herrmann (2016). "Job Shop Scheduling With Flexible Energy Prices", ECMS 2016 Proceedings edited by: Thorsten Claus, Frank Herrmann, Michael Manitz, Oliver Rose European Council for Modeling and Simulation. doi:10.7148/2016-0488
[2] Hubert Hadera, Iiro Harjunkoski, Guido Sand, Ignacio E. Grossmann, and Sebastian Engell. "Optimization of steel production scheduling with complex time-sensitive electricity cost". Computers & Chemical Engineering, 76(0):117 – 136, 2015.
[3] A. Haït and C. Artigues. "A hybrid CP/MILP method for scheduling with energy costs". European Journal of Industrial Engineering, 5(4):471–489, 2011.
[4] Hooker, J. N. "Planning and Scheduling by Logic-Based Benders Decomposition." Operations Research, vol. 55, no. 3, 2007, pp. 588–602.

Diploma Thesis Supervisor: Ing. István Módos

Valid until the end of the summer semester of academic year 2017/2018

L.S.

prof. Dr. Michal Pěchouček, MSc.
Head of Department

prof. Ing. Pavel Ripka,CSc.
Dean

Prague, February 15, 2017

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Science and Engineering

Master's Thesis

Scheduling of energy-demanding operations with varying cost of electricity

Bc. David Král

Supervisor: Ing. István Módos

Study Programme: Open Informatics

Field of Study: Software Engineering

May 19, 2017

Aknowledgements

I would like to thank my supervisor Ing. István Módos and Ing. Přemysl Šůcha, Ph.D. for guidance, valuable advices and regularly provided consultations. I would also like to thank my family and my girlfriend for supporting me during my studies.

Declaration

I hereby declare that I have completed this thesis independently and that I have listed all the literature and publications used.
I have no objection to usage of this work in compliance with the act §60 Zákon č. 121/2000Sb. (copyright law), and with the rights connected with the copyright act including the changes in the act.

In Prague on May 25, 2017 ....

Abstract

This thesis deals with the scheduling of highly energy-demanding jobs to identical parallel machines considering a varying price of consumed electric energy, which is an NP-hard combinatorial problem. It typically occurs at manufacturing process as melting of steel or glass hardening. Typically, the manufacturing company and the supplier of the electric energy have a pre-agreed contract according to which the price of consumed electricity is charged. For the long-term contract, the price is constant for a long period, but there exist contracts, where the price is varying during much shorter time periods (e.g. hours). To minimize the price for consuming electricity, automated computer approaches that try to intelligently schedule the jobs could be use. This can be achieved by shifting the highly energy-demanding jobs into the time periods, in which the price for consuming the electricity is low.

Four exact methods and two heuristics for solving the described scheduling problem are introduced by this work. Two of the exact methods are fully our contribution and the remaining two methods are slightly modified approaches found in the state-of-art literature and primarily used for comparison. The procedure for generating a lot of distinct test instances is also proposed in this work. The generated instances were used for performing thorough experiments, which verify the effectiveness and accuracy of the designed methods.

Abstrakt

Tato práce se zabývá rozvrhováním energeticky vysoce náročných prací na paralelní identické stroje v závislosti na proměnné ceně za spotřebovanou elektrickou energii, což je NP-těžký kombinatorický problém. Ten typicky nastává při výrobním procesu, jakým je například kalení oceli či tvrzení skla. Výrobní společnost má s dodavatelem energie určitý předem uzavřený kontrakt, dle kterého je jí účtována cena za spotřebovanou elektrickou energii. Při dlouhodobém kontraktu je cena po dlouhou časovou periodu konstantní, ale existují i takové kontrakty, kde se cena mění během mnohem kratších časových period (např. hodin). Pro dosažení minimalizace ceny za spotřebovanou energii by mohli být použity automatizované počítačové přístupy, které se snaží práce inteligentně rozvrhovat. Minimalizace ak může být dosaženo například posouváním vysoce energeticky náročných prací do časových period s nízkou cenou energie.

V práci jsou představeny čtyři exaktní a dvě heuristické metody pro řešení popsaného problému. Dvě exaktní metody jsou naším příspěvkem a dvě, sloužící především pro porovnání, jsou převzaté z literatury a upraveny tak, aby dokázaly náš rozvrhovací problém řešit. Tato práce dále obsahuje postup pro generování velkého množství rozličných testovacích instancí, které byly použity pro podrobení všech navržených metod důkladným experimentům ověřujících jejich a efektivnost a přesnost.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

An increase of energy cost by 130% was noted in an industrial sector between the years 2000 and 2010 according to [19]. The industrial sector is the largest energy consumer in the world and energy consumption by this sector has doubled in the past 50 years. Most of the energy consumed by production enterprises is in the form of electricity [7].

Nowadays, an electricity market allows customers to buy an electric energy for the prices that are continuously changing during a day and that are known some time ahead (i.e. *day-ahead market*) instead of prices that are constant during a longer period (e.g. month). Purchasing on the mentioned day-ahead market, which is one of the types of the short-term electricity market, is very common in Europe [20]. In the Czech Republic, the amount of traded energy on the day-ahead market was 19.97 TWh in 2015 [17]. For an illustration how the price can vary during the day, the Figure 1.1 is attached. Note that in some periods, the price of electric energy is much lower than in periods, in which the price is very high.

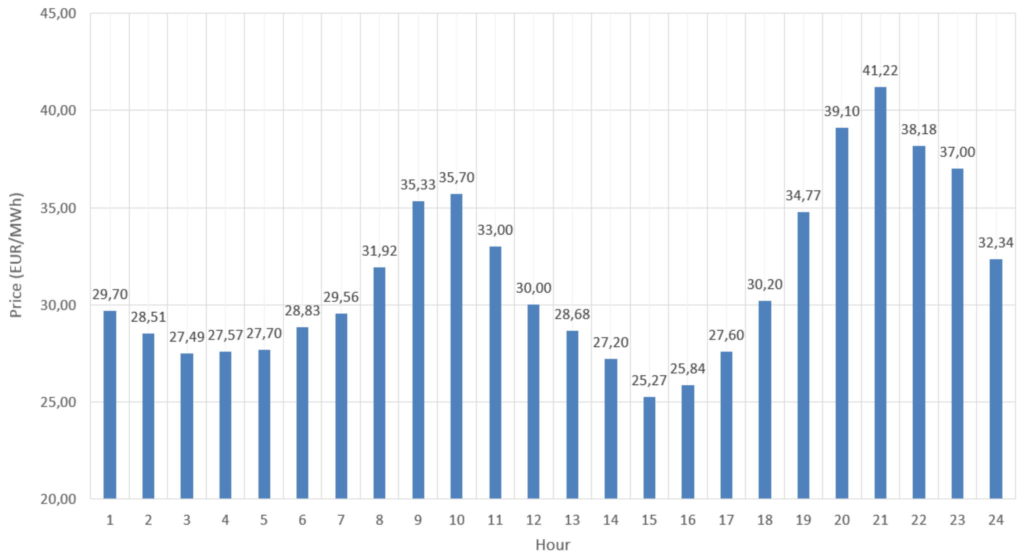
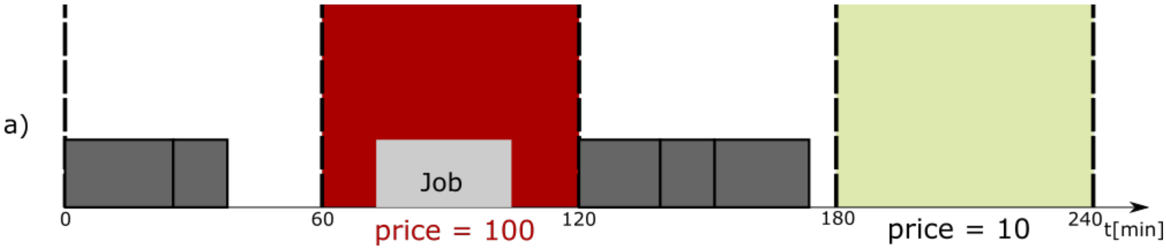
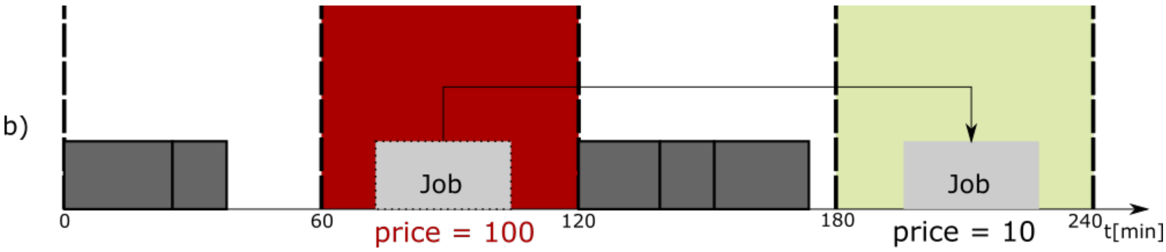


Figure 1.1: Hourly varying electricity prices in the day-ahead market 1.4.2017 [16]

In connection with an existence of the energy-demanding industrial processes, a big potential arises to achieve significant savings in the expenses for energy. For instance, a factory can schedule the most energy-demanding operations into time periods, in which the price of electricity is lower than in the others. In combination with the case that the factory does not need to produce their products as soon as possible, the described shifting of the operations within the allowed time window to cheaper time periods can lead to minimizing the electricity cost. The time overlap with time periods with defined electricity price influences the electricity consumption cost for processing the operation (see Figure 1.2).



(a) Job processed in expensive period



(b) Job shifted into the cheaper period

Figure 1.2: Shifting job into cheaper period

According to the previous paragraphs, the aim of this work is to design and construct algorithms, which schedules jobs on identical parallel machines and ensures that the total electricity cost is as minimal as possible. The jobs represent the operations of industry processes. Four exact methods together with heuristics were designed, and effectiveness and correctness of all the methods were experimentally tested.

1.1 Situation on an electricity market

The trading with the electric power is unique, primarily because of its difficult storability and because of the equality between an amount of produced and consumed energy (together with losses) that needs to be respected in real-time [9]. ČEPS, a. s. cares for the balance in the Czech Republic. Many parties participate in an electricity market, e.g. producers of energy, traders, consumers, operator of a transmission system, operators of a distribution network, etc. [23]. One of the participants is an operator of the market OTE, a. s. whose responsibility is to evaluate the offers and demands of the electricity and convey it to the operator of the transmission system and operators of the distribution network. Another responsibility is to evaluate the long-term perspective of the offers and demands and gives the information to the participants of the market. The most important fact is that operator cares for organizing of a short-term (day-ahead, spot, block, etc.) electricity market, which is an inseparable part of the problem addressed in this work. The short-term electricity market is according to [17] among the others divided into these types:

- Block market: trading with the block products for a given day
  - Base: whole day
  - Peak: 8:00 - 20:00
  - Off-Peak: 0:00 - 8:00 and 20:00-24:00

  The trades are always concluded few days before the day of delivery.

- Day-ahead market: it is possible to anonymously demand and offer the electricity for every hour of given day and the price is determined based on the analysis of the offers and demands. The trading on this type of market results to established amounts of traded electricity for every hour of given trading day.

- Spot market (intra-day): Trading, which takes place minimally one hour ahead in given day. The traders anonymously offer and demand the electricity for a specific hour of given day.

These days, a large effort is invested on getting a power grid to be more stable, efficient and reliable. Specifically, a balance between an amount of created and consumed electric energy is needed to be fulfilled. In a case of surplus or a deficit of the consumed and generated energy, the quality of transmitted electricity is negatively affected. It is not trivial to fulfill these requirements due to significant changes in the consumed amount of electric energy during the day and because of greater involvement of renewable resources whose availability is highly variable. The short-term electricity market is organized also for the described reasons and it helps to the operator of the grid to attain that the energy is consumed according to the power grid needs, e.g. the load curve can become more flattened and the required amount of energy generated during the peak periods can be reduced [24].

1.2 Scheduling

Because of the existence of the short-term electricity market, in which the price can vary very often, the consumers can optimize their business positions even at a short time before the energy delivery time (e.g., day or hour). They can react to the actual situation in their production or consumption portfolio[17] and appropriately schedule their processes. According to [4], the scheduling solves a problem, when sets of jobs and machines are given, and some assignment of the jobs to the machines has to be done. The goal of the scheduling is to create a schedule, which assigns every job to some machine for a fixed time duration and which also satisfies other requirements and constraints. An instance of schedule is presented in Figure 1.3, illustrates that creating a schedule is not a trivial task due to large space of possible combinations.

For every job, the machine, which has to be allocated to it, may be given in advance. This type of the machines is denoted as *dedicated*. In case that every machine from the set of

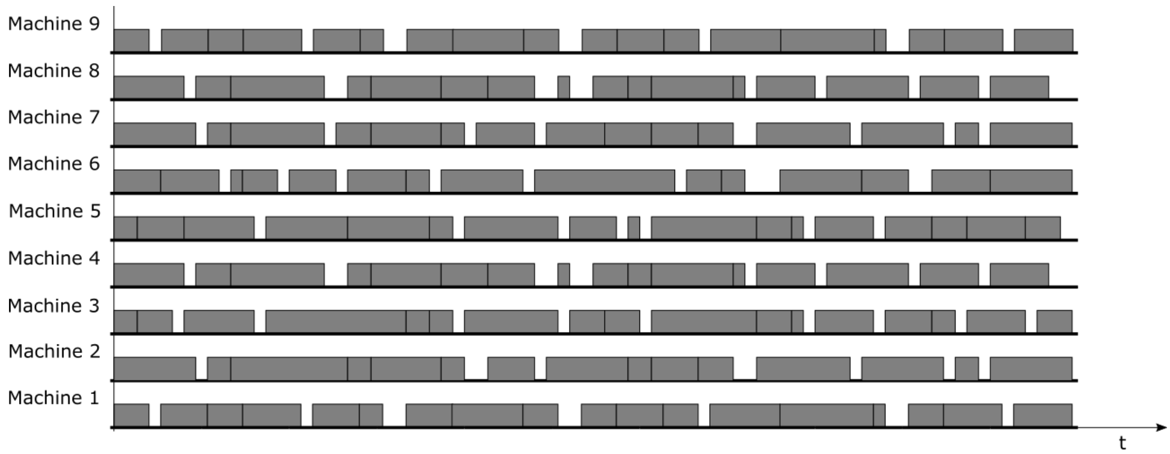


Figure 1.3: Example of a schedule

the machines can be chosen for performing the job, we denote the machines as *parallel*. The parallel machines are divided into subtypes according to their parameters. Some of them can have different parameters (e.g. production speed, electricity consumption, etc.) or they can be identical.

Every job is restricted by some parameters, for example, the minimal time, when an execution of the job can be started (release time), duration of a processing of the job (processing time) or the maximal time until the job has to be finished (deadline). The schedule that satisfies these restrictive parameters of the jobs and in which no two jobs are overlapping on any machine is denoted as *feasible* schedule. However, in most cases, a set of another constraint is given, and the feasible schedule must respect them all.

Generally, finding only the feasible schedule does not have to be sufficient. In most cases, we want to find schedule that is efficient, e.g. we don't want to waste our resources such as time, electricity, money, etc. For this reason, another goal of the scheduling is to find the schedule, which minimizes or maximizes some optimality criterium. A large number of optimality criteria exists in both practice and literature [4]. Here, some of them are listed:

- makespan: completion time of the last scheduled job
- lateness: the difference between the time of completion of a job and its deadline
- price of the schedule: the price can be given in the form of an arbitrary function

In our case, some parameters of jobs or machines linked with electricity consumption have to be defined. Together with the defined price of electricity, the optimality criterium in the form of the price of the schedule can be defined as the overall cost of the electricity consumption.

In principle, the scheduling problems can be solved by two types of the methods. First of them are the exact methods, which always find an optimal solution, if it exists. One of these methods is Mixed Integer Linear Programming (MILP), which is described in [18]. The method is based on modelling of the combinatorial problems (e.g. scheduling) using linear expressions with both integer and continuous variables. The algorithms for MILP then try to find a feasible values for the variables, i.e. such values that satisfy the linear constraints. The MILP method is used for a mathematical notation of the designed exact methods for solving the scheduling problem addressed in this work. The disadvantage of the exact methods is that the searching for the optimum can take a very long time (generally, solving the MILP is NP-hard problem) and that's not acceptable for solving the scheduling problems in real situations. For this reason, methods called heuristics are developed, which find suboptimal solutions in a fraction of the time needed by an exact method.

1.3 Related work

Due to the conditions mentioned in the previous section, there is in last years an increased amount of an effort given into a research area dealing with the scheduling of the industry processes concerning the varying prices of a consumed unit of the electric energy. Some of these industry processes occur in steel manufacturing, which is even regarded as the one of the most energy demanding production according to [24] and many of the studies deal with the scheduling of them. There is a lot of specific technical constraints in steel manufacturing that need to be respected, and that can vary from company to company. Most of the studies goal is to minimize the time required to produce a given amount of products, but there is an increased number of works dealing with reducing the costs linked with consumed electric energy by production. One of these studies is [12], in which the emphasis is on minimizing idle times, where energy is consumed, but nothing is produced. These idle times can, for instance, be caused by an insufficient number of operators that controls the production. It is also necessary not to exceed the amount of the electric energy agreed with the supplier to be consumed. The given scheduling problem was by the authors of the research transformed to the Parallel Machine Scheduling Problem that is the problem of the scheduling of tasks on the parallel machines described in section 1.2. The authors designed a mathematical model, which can identify start times of all phases of the tasks and allocate the machines to them. But there is a problem that this designed procedure is not able to solve larger instances inspired by real production. So the research is extended by a two-phase heuristic method based on the decomposition of the problem into two phases, which are repeated until the outputs of both phases are not the same, and the schedule is feasible.

In principle, it is possible to create two types of the mathematical models: discrete-time model and continuous-time model. In the discrete-time model, the scheduling horizon is divided into the uniform time periods, and the variables are indicating if there is some event starting in given period (time-indexed model). In the continuous-time model, the variables indicate start time or end time of some event, so the number of variables is reduced. Both types of the models used in the scheduling of the industrial processes are compared in [24]. The model used in previously described research [12] is continuous-time, and it is based on a new formulation, focused on the relative positions of tasks and time periods. This type of model is used in another research [13], where it is compared with different continuous-time formulation previously developed in other work.

Next research focused on the scheduling with the goal of minimizing the costs linked with the electricity consumption is [11]. There, the previously published continuous-time MILP model with precedences is extended, and the extension relates with the depending on the amount of the consumed electricity. In a single step, the makespan and the costs of consumed energy are minimized, what is very time-consuming, so the described method is limited only to small instances of the problem. The authors dealt with it by using the multilevel heuristic algorithm. In the research, three different contracts with energy supplier are considered. The first two are related to short-term electricity market described in section 1.1. The first of them is trading on the day-ahead market, the second contract is trading on the block market, and the third is the long-term contract, where the price of the electricity is constant for an agreed time horizon. Also, the producing of energy by a company and the option to sell it back to the grid related with it is considered. The finding of the solution includes the identification of the minimum-cost flow in an imaginary graph, where all three described contracts and the producing of electricity are considered.

In the research [22], the scheduling focused on minimizing the costs of the consumed electricity by production is divided into two phases. Firstly, individual tasks are partitioned into groups according to similar properties of products produced by that tasks. In the first step, the model, which identifies the start times of individual tasks of individual groups is solved. And in the second, whole groups are scheduled, and the problem is transformed to the Resource-Constrained Project Scheduling that is the problem of scheduling the task groups, for which some minimal amount of needed resources is given, and the schedule has to respect the capacities of that resources, which can not be exceeded. In this phase, the price of energy, which is determined by trading on block market, is considered. Another related work, which examines trading on block market is [1]. It deals with shifting the production from peak intervals with high energy price to other intervals, in which the price of electricity is lower. Also, the constraints linked with a storing the products are considered, and the situation in India, where still the most of the steel-making companies follow the long-term contracts with energy suppliers, is described. The consuming of energy in peak intervals can be reduced by 50% according to results of the research.

The scheduling considering the price of the electricity traded on the day-ahead market is addressed in research [20]. In work, the authors focus on the scheduling the tasks on the single machine. They show that by the scheduling of appropriate states of the machine to appropriate time periods the considerable savings can be done in the costs of energy. A state diagram is described, which contains three possible states of the machine: Processing, Idle (turned on, but does not process) and Shut-down. The edges between states Shut-down and Processing are marked by times and consumptions of the energy that are needed to transition between these states. The pre-emption (partly processing a task and finishing it after processing another task) is not allowed in the addressed problem. The goal is to schedule the operations when the machine is in the Processing state to periods, where the price is lower and on the other hand keep the machine in the other states in more expensive periods. For solving larger instances, the genetic algorithm is introduced, and outputs of this algorithm and the analytical method are compared.

Another two works use Resource-Task Network (RTN) as a representation of production processes. First of them is [5], in which two distinct models are introduced: discrete-time and continuous-time. The continuous-time model was first of its type that was able to work with the varying price of energy. Also, the authors focused on forecasting the demands of products using stocks linked with it. The time horizon for scheduling is one week, and the energy is purchased on the block market, meaning that price can take on of the three values. It is shown that the continuous-time model can solve only small instances of the problem. In opposite, the discrete-time can process more efficiently, but with bigger divergences from the results produced by analytical methods and it is primarily suitable for decision making because of this reason. The second and the newer work is [6], in which the new aggregate discrete-time formulation is described, which is combined with another continuous-time model, and together, it forms the Rolling-Horizon algorithm. The aggregate formulation merges the time periods that has the same price of energy to a single period. The results of the Rolling-Horizon algorithm are compared with the results of two formulations from the previous work.

To test the effectiveness of various mathematical models and algorithms dealing with the scheduling of industrial processes attempting to minimize the costs of consumed electric energy not only with randomly generated instances of the problem, the research [10] was performed. It describes the implementation and deployment of one concrete method dealing with described scheduling problem in a real factory. The continuous-time MILP model was developed, which is used in a heuristic method that is based on a decomposition of the problem into smaller subproblems, and the solution is found in shorter time. At first, the tasks producing similar products are divided into groups, which are optimized as a Job-Shop Problem, where every task (job) consists of several operations, and for all the operations, the machine, which has to be allocated to them is given. Next step is to schedule every group as a Flow-Shop Problem, which differs from more general Job-Shop Problem in that the predetermined order of performing the operations must be respected. The output of described decomposition is the schedule, which minimizes the makespan. In the last phase, another mathematical model is added, and the price of energy is included in it. After the year of usage described method, the factory created 3% savings in costs of electric power, it became more flexible and adaptable to unexpected events, and the coordination between distinct stages of production improved.

The studies presented so far dealt with the scheduling of the processes of steelmaking. But many studies are dealing with scheduling of another energy highly consuming industrial processes. One of these studies is [15], in which two discrete-time models for minimizing the costs of energy consumption are developed. The first of them deals with the scheduling the processes of air separation plant and the second deals with the processes of cement manufacturing plant. A lot of distinct methods of DSM considering many industrial areas are described in [24].

The important thing is that the studies presented so far deal with very concrete problems with very concrete constraints, so the solution space is significantly reduced. For instance, in steelmaking, the order of performing the operations of tasks is prescribed, and it needs to be followed. Another instance of reducing the solution space can be that many of steelmaking plants have a very limited count of parallel machines in individual phases of production (e.g. one or two). Because of that, the described algorithms, which can solve that problems in a short time are not able to solve more general problems efficiently without the use of some heuristic method. One of the few studies that deal with more general scheduling problems and doesn't use the heuristic methods is [19]. The authors focus on solving the previously described Job-Shop Problem with the goal to minimize the costs of electricity traded on the day-ahead market. A discrete-time model is introduced, which schedules the operations of jobs to the time periods and the states of the machines depending on the schedule are chosen. It is important that the model can choose, if the machines will turn off in downtimes between the performing of the operations, or if the machines will stay in the Idle state. The problem is that the described method works efficiently only with the scheduling horizon divided into whole hours. For the problems occurring in real industries, the discretization to much shorter time units is more appropriate. For this fine granularity, the described method is not able to solve the problems in short time enough.

Another work is [7], where the parallel machine scheduling problem with Time-of-Use (TOU) electricity prices is addressed. TOU prices are another title for prices descended from trading on block electricity market. The parallel machines are unrelated, meaning that some of them have different processing speed and higher energy consumption related to it. The authors presented exact MILP method for minimizing total energy cost. The method exploits the fact that the prices of electricity remain steady for relatively long periods (e.g. 5 to 8 hours) and the objective is unchanged when the job is shifted within the same period. The resultant schedule must respect some given production deadline and no release times of jobs are considered. The problem of the method is that the performance decreases significantly when the price of electricity frequently vary (every hour/half hour). For this reason, a heuristic method based on decomposition is presented. Firstly, the jobs are assigned to the machines, and then, every machine is scheduled separately. The objective is to minimize the makespan together with minimizing the energy cost.

1.4 Contribution of the thesis

The previous sections show that there is no exact efficient method yet for solving relatively general scheduling problem considering the varying price of electricity with the goal to minimize the costs of electricity consumption, which is NP-hard in the strong sense as proved in Section 2.4. The methods designed in this work allocate the identical parallel machines for time-limited jobs (with release times and deadlines), which can have different processing times and electricity consumptions. Two exact methods are fully our contribution. The remaining two exact methods are approaches inspired by the literature and modified to fit our problem, and they are used primarily for a comparison. The output of the exact methods is a schedule, which ensures that the cost of electricity consumption is as minimal as possible. The proposed algorithms can work with the price of electricity that can change itself very often (e.g. every hour) so that the power could be purchased on short-term market described in section 1.1. The significant contribution of this work is that the values of processing times and starting times of the jobs do not have to be only whole hours as it is in research [19]. We also designed heuristics that incorporate our efficient exact algorithms to solve smaller parts of the whole problem (*matheuristics*) for cases, where the exact methods are not able to find solutions in acceptable time.

1.5 Outline of the thesis

In next Chapter 2, the problem of the scheduling of time-limited jobs with different processing times on the identical parallel machines with the goal to minimize the cost of the electricity consumption is described. That is followed by Chapter 3, where the design of four exact methods solving the described problem is presented, including the description of all the steps needed to be done for finding the solution. The heuristic methods for finding an acceptable solution in short time are presented in Chapter 4. The results of the experiments that examine the effectiveness and correctness of exact methods together with heuristics on a large number of different instances of the problem are presented in Chapter 5. The last Chapter 6 concludes the work.

Chapter 2

Problem statement and complexity

In Section 2.1 the problem of the scheduling of jobs on identical parallel machines is described. In Section 2.2 the problem of scheduling the jobs with the goal to minimize the cost of energy consumption of all machines is presented. In Section 2.3, the whole problem addressed in this work is stated.

2.1 Scheduling on identical parallel machines

Let J be a set of jobs and M be a set of identical parallel machines. For each $j \in J$ we define:

- Release time $r_j \in \mathbb{N}_0$: time, when job j is ready to be processed
- Processing time $p_j \in \mathbb{N}_{>0}$: duration of the processing of job j
- Deadline $d_j \in \mathbb{N}_0$: time, until job j has to be processed

The goal is to find a schedule, i.e. start time and machine allocated for every $j \in J$, such that all constraints are satisfied, i.e. no deadline is exceeded, and no job starts before its release time. The machines are identical and parallel, meaning that every machine $m \in M$ has the same processing speed and it can be allocated for every job $j \in J$. Note that the start time of job j has to be from the interval $[r_j, d_j - p_j]$

2.2 Variable electricity price

In our problem, the scheduling horizon is divided into a set N of uniform time periods denoted as n. A length of all the periods is denoted as $D \in \mathbb{N}_{>0}$. The length of scheduling horizon is denoted as $\Gamma \in \mathbb{N}_{>0}$. Due to the practical reasons, the length of all the periods is set to one hour.

For each period $n \in N$ we define a price $price_n$ for consuming the energy in it. The price can be obtained from the short-term electricity market described in Section 1.1.

The processing of every job $j \in J$ consumes some amount of electric energy. The overall cost of consumption of j is in some schedule depends on:

- power rate c_j needed by job j
- processing time p_j of job j
- job/time overlaps in a schedule with time periods $n \in N$, for which the price for consuming energy in it is in advance
- price $price_n$ of period $n \in N$

For an illustration how the cost of consumption of job j can vary with its start time, the figure 2.1 is attached.

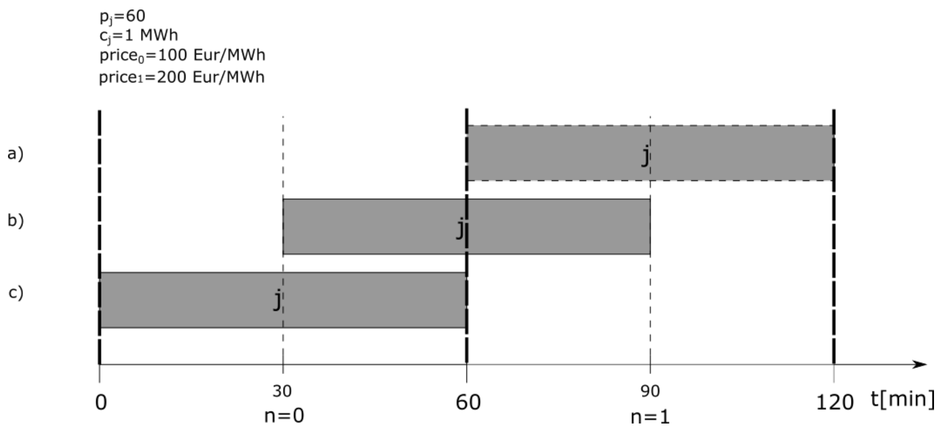


Figure 2.1: Varying costs of electricity consumption depending on three different start times

Based on different job/period overlaps $o_{j,n} \in [0, D]$, the costs of processing the job j depending on its start time are counted by this formula

$$\frac{o_{j,0} \cdot c_j \cdot price_0 + o_{j,1} \cdot c_j \cdot price_1}{60}$$

The costs depending on three different values of start time of job j are:

- a): 200 Euro
- b): 150 Euro
- c): 100 Euro

2.3 Problem statement

Now, we can extend the scheduling problem from the section 2.1 by the goal to create a schedule, in which the energy consumption costs are minimized resulting in the **Identical Parallel Machine Scheduling Problem Minimizing Energy Costs** problem, which will be denoted as **IPMSMEC** by now.

2.4 Problem complexity

The aim of this section is to show that the IPMSMEC problem addressed by this work is NP-hard in the strong sense. Although the problem of scheduling jobs with release times and deadlines on one machine is already NP-hard according to [4], we show that the scheduling of jobs to time periods with the defined price of electricity on one machine is NP-hard too. In the proof, we will use the decision version of the scheduling problem under varying prices, i.e. is there a schedule which has a cost of consumed electricity equal or less than Q?

Theorem: The problem of scheduling the jobs (even without release times and deadlines) under varying electricity price is NP-hard in the strong sense.

Proof: We can use a reduction from a strongly NP-complete problem 3-Partition [8] described as follows. A positive integer bound B is given together with set A of $3m$ elements, where every element $x \in A$ has a positive integer size $s(x)$ that satisfies $B/4 < s(x) < B/2$ and the sum of the sizes of elements in A is exactly mB. The question is, if A can be partitioned into m disjoint sets $A_1, \ldots, A_m$ such that each A_i contains exactly 3 elements of A and each A_i has total size equal to B.

Now, we can set a cost of consumed electricity by a schedule for our decision problem to $Q = 0$. The instance of the described 3-Partition decision problem can be transformed into the instance of our decision problem as follows. For every element $x \in A$ with size $s(x)$ we construct job $j \in J$ with $p_j = s(x)$ so that $A = J$. After that, we can construct a set of time periods $n \in N$, so that $|N| = 2m$. The length of every period $n \in N$ noted as D is set to bound B. The price of electricity in period n is $price_n = \begin{cases} 0, & \text{if n is even} \\ 1, & \text{if n is odd} \end{cases}$

The answer to 3-Partition decision problem is *yes* if and only if the answer to our decision problem of creating a schedule, for which the resulting cost of electricity consumption is $Q = 0$ *yes*. If the answer was *yes*, there was a schedule, in which we did not have to schedule some job to period with $price_n = 1$. It means that we were able to divide a set J into the disjoint triples of jobs $j \in J$ so that all triples have the same sum of processing times equal to D. After that, we could schedule the triples of jobs to all the periods with electricity price equal to 0 so that the cost of the resulting schedule is 0. The triples of jobs exactly match to the triples of elements in A for 3-Partition decision problem and all the triples have total size equal to B.

For an illustration, the Figure 2.2 is attached, where jobs $j \in 0 \ldots 5$ are scheduled.

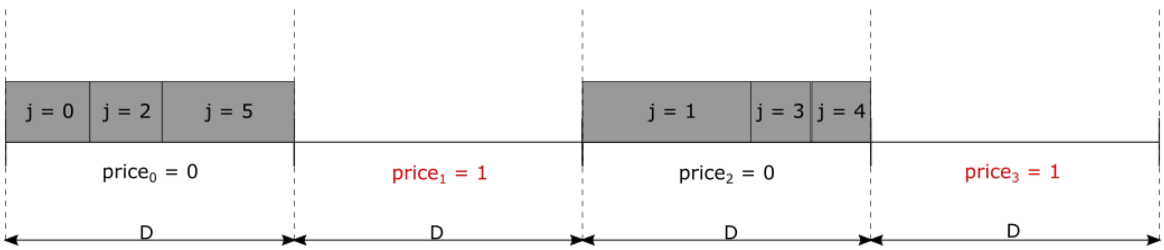


Figure 2.2: 3-Partition problem reduction

Chapter 3

Design of exact methods

In this chapter, four exact methods for solving the previously described IPMSMEC problem are presented. First two of the methods are standard formulations from the literature slightly modified to fit our problem. The proposed exact methods are:

- Continuous MILP formulation, see Section 3.1
- Time-indexed MILP formulation, see Section 3.2
- Two-phase method, see Section 3.3
- Logic-Based Benders Decomposition, see Section 3.4

3.1 Continuous formulation

The first designed method for the IPMSMEC problem is the continuous-time MILP formulation based on the formulation presented in [12]. It has been slightly modified to fit our problem.

The formulation is based on fixing the values of a continuous variable defining start time for every job. Based on the start time, we can determine the exact overlap o_j, m, n with every time period on every machine. After that, it is possible to allocate the machines for the jobs with avoiding of processing more jobs simultaneously on one machine, and it is possible to determine the exact energy consumption cost, which has to be as minimized as possible.

3.1.1 Mathematical model

Sets:

J	Set of jobs	$j \in J$
N	Set of uniform time periods	$n \in N$
M	Set of machines	$m \in M$

Parameters:

m	Number of machines
$price_n$	Price of energy in period n
D	Length of time periods
Γ	The scheduling time horizons length
c_j	Electricity consumption of job j
p_j	Processing time of job j
r_j	Release-time of job$j$
d_j	Deadline of job j

Variables:

$t^s_{j,m}$ Start-time of job j on machine m

$t^f_{j,m}$ Finish-time of job j on machine m

$$y_{j,m} = \begin{cases} 1, \text{if job } j \text{ is processed by machine } m \\ 0, \text{else} \end{cases}$$

$$v_{j,j',m} = \begin{cases} 1, \text{if job } j \text{ is scheduled on machine } m \text{ before job } j' \\ 0, \text{else} \end{cases}$$

$$\gamma^s_{j,m,n} = \begin{cases} 1, \text{if job j } j \text{ on machine } m \text{ starts before or during the period } n \\ 0, \text{else} \end{cases}$$

$$\gamma^f_{j,m,n} = \begin{cases} 1, \text{if job j } j \text{ on machine } m \text{ ends before or during the period } n \\ 0, \text{else} \end{cases}$$

$o_{j,m,n}$ Time overlap of job j on machine m with period n

Objective

$$\min \sum_{j\in J}\sum_{m\in M}\sum_{N\in n} c_j \cdot price_n \cdot o_{j,m,n}/D \tag{3.1}$$

subject to

$$t^s_{j,m} = t^f_{j,m} - p_j \cdot y_{j,m} \qquad \forall j \in J, m \in M \tag{3.2}$$

$$r_j \cdot y_{j,m} \leq t^s_{j,m} \qquad \forall j \in J, m \in M \tag{3.3}$$

$$d_j \cdot y_{j,m} \geq t^f_{j,m} \qquad \forall j \in J, m \in M \tag{3.4}$$

$$\sum_{m\in M} y_{j,m} = 1 \qquad \forall j \in J \tag{3.5}$$

$$v_{j,j',m} + v_{j',j,m} = 1 \qquad \forall (j,j') \in J^2, m \in M \tag{3.6}$$

$$t^f_{j,m} \leq t^s_{j',m} + \Gamma(1 - v_{j,j',m}) \qquad \forall (j,j') \in J^2, m \in M \tag{3.7}$$

$$t^f_{j',m} \leq t^s_{j,m} + \Gamma v_{j,j',m} \qquad \forall (j,j') \in J^2, m \in M \tag{3.8}$$

$$t^s_{j,m} \geq D \cdot (n+1)(1 - \gamma^s_{j,m,n}) \qquad \forall j \in J, m \in M, n \in N \tag{3.9}$$

$$t^s_{j,m} \leq D \cdot (n+1) + \Gamma(1 - \gamma^s_{j,m,n}) \qquad \forall j \in J, m \in M, n \in N \tag{3.10}$$

$$\gamma^s_{j,m,n+1} \geq \gamma^s_{j,m,n} \qquad \forall j \in J, m \in M, n \in N \tag{3.11}$$

$$\gamma^s_{j,m,|N-1|} = 1 \qquad \forall j \in J, m \in M \tag{3.12}$$

$$t^f_{j,m} \geq D \cdot (n+1)(1 - \gamma^f_{j,m,n}) \qquad \forall j \in J, m \in M, n \in N \tag{3.13}$$

$$t^f_{j,m} \leq D \cdot (n+1) + \Gamma(1 - \gamma^f_{j,m,n}) \qquad \forall j \in J, m \in M, n \in N \tag{3.14}$$

$$\gamma^f_{j,m,n+1} \geq \gamma^f_{j,m,n} \qquad \forall j \in J, m \in M, n \in N \tag{3.15}$$

$$\gamma^f_{j,m,|N-1|} = 1 \qquad \forall j \in J, m \in M \tag{3.16}$$

$$0 \leq o_{j,m,n} \leq D(\gamma^s_{j,m,n} - \gamma^f_{j,m,n-1}) \qquad \forall j \in J, m \in M, n \in N \tag{3.17}$$

$$o_{j,m,n} \geq D \cdot (\gamma^s_{j,m,n-1} - \gamma^f_{j,m,n}) \qquad \forall j \in J, m \in M, n \in N \tag{3.18}$$

$$o_{j,m,n} \geq t^f_{j,m} - D \cdot (n+1) + D \cdot \gamma^s_{j,m,n-1} - \Gamma(1 - \gamma^f_{j,m,n}) \qquad \forall j \in J, m \in M, n \in N \tag{3.19}$$

$$o_{j,m,n} \geq D \cdot (n+1)(1 - \gamma^s_{j,m,n-1}) - t^s_{j,m} - D \cdot \gamma^f_{j,m,n} \qquad \forall j \in J, m \in M, n \in N \tag{3.20}$$

$$\sum_{n\in N} o_{j,m,n} = p_j \cdot y_{j,m} \qquad \forall j \in J, m \in M \tag{3.21}$$

The equation (3.2) fixes the finish time of job j to be equal to the start time together with the processing time in case that the job j is assigned to be processed on machine m. Equations (3.3) – (3.4) require that the job j will be processed on machine m in the time window prescribed by the release time and deadline. Constraint (3.5) ensures that exactly one machine is allocated for job j. The equations (3.6) – (3.8) guarantee that at most one job is processed at a time on every machine. The following (3.9) – (3.12) constraints use and fix variable $\gamma^s_{j,m,n}$ to indicate whether the job j starts before or during the interval n on machine m. Variable $\gamma^f_{j,m,n}$ is fixed in the same manner by constraints (3.13) – (3.16).

The equations (3.17) – (3.21) define the overlaps of jobs with time periods based on start times and end times of the jobs. The sum of all job/period overlaps has to be equal to the jobs processing time. The authors of [12] describe that there are six possible configurations of a relative position of the job and the period and the type of the configuration have to be determined to get the overlap $o_{j,m,n}$. For an illustration of all possible job/period position configurations, see Figure 3.1.

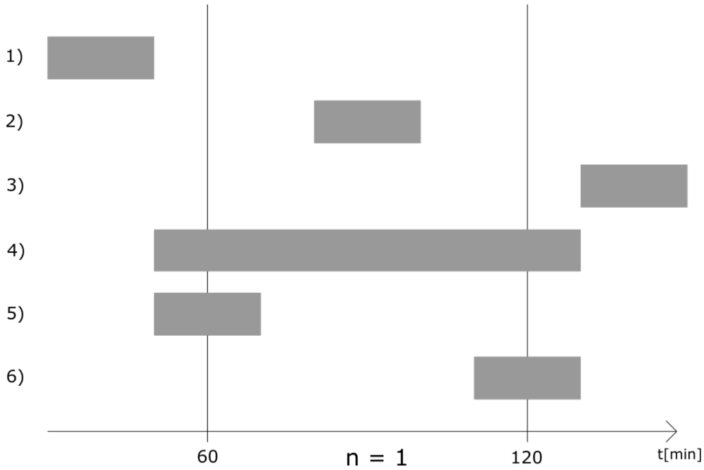


Figure 3.1: Six possible job/period position configurations

It is important that the overlap $o_{j,m,n}$ will be greater than zero only if $\gamma^s_{j,m,n} - \gamma^f_{j,m,n-1} = 1$ as mentioned and presented in [13]. For an illustration, how the connection between $o_{j,m,n}$, $\gamma^s_{j,m,n}$ and $\gamma^f_{j,m,n}$ works, the Figure 3.2 is attached. When the overlaps of all jobs with the time periods are known, it is possible to define the objective given by equation (3.1), which is to minimize the overall electricity consumption costs.

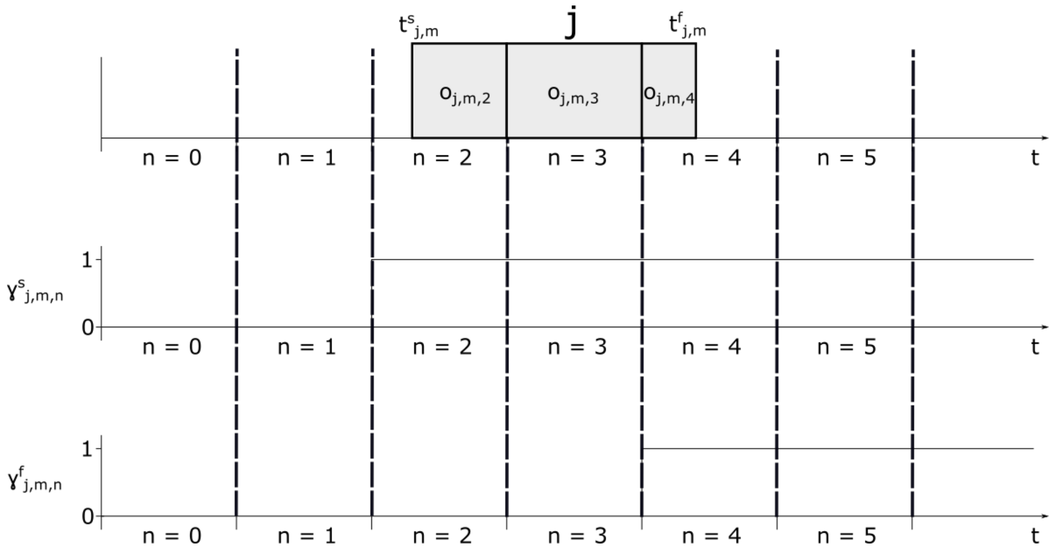


Figure 3.2: Job/period overlaps definition

3.2 Time-indexed formulation

The second MILP formulation is a time-indexed one, where the variables are binary and indexed by a time, and that indicate if some event starts at the given time point. The formulation defines the start times of all jobs together with the allocation of machines to them in one step. The model uses a binary variable $s_{j,t,m}$, which indicates whether job j starts at time t on machine t or not.

3.2.1 Preprocessing

Time window of possible start times for job $j \in J$ can be identified with knowledge of release time and deadline. This increases the efficiency since the space of possible start times is pruned. The time window is represented by a set $T_j \subseteq T$, where T is a set of all time points of scheduling horizon. The number of $s_{j,t}$ variables is reduced because of because of computing the time windows. The set T_j contains all the time points from interval

$$[r_j, d_j - p_j]$$

Note that a price $price_{j,t}$ for consuming an energy by a processing a job starting at some time t can be computed in advance. The figure 2.1 shows, how the price is affected by the job/period overlaps, processing time and the electricity consumption of the job and by the price $price_n$ for consuming energy in the period, to where the time point t belongs.

3.2.2 Mathematical model

Sets:

J	Set of jobs $j \in J$
T	Set of uniform points in time $t \in T$
T_j	Set of uniform points in time, in which job j can start
M	Set of machines $m \in M$
N	Set of uniform time periods $n \in N$

Parameters:

m	Number of machines
$price_n$	Price of energy in period n
$price_{j,t}$	Price of consuming an energy by a processing job j starting at time t
c_j	Electricity consumption of job j
p_j	Processing time of job j
r_j	Release-time of job j
d_j	Deadline of job j

Variables:

$$s_{j,t,m} = \begin{cases} 1, \text{if job } j \text{ starts at time } t \text{ on machine } m \\ 0, \text{else} \end{cases}$$

Objective

$$\min \sum_{j \in J} \sum_{t \in T} \sum_{m \in M} price_{j,t} \cdot s_{j,t,m} \tag{3.22}$$

subject to

$$\sum_{t \in T} \sum_{m \in M} s_{j,t,m} = 1 \qquad \forall j \in J \tag{3.23}$$

$$\sum_{j \in J} \sum_{k=max(0,t-p_j+1)}^{t} s_{j,k,m} \leq 1 \qquad \forall t \in T, m \in M \tag{3.24}$$

Equation (3.23) ensures that every job starts at most at one time on one machine from set M. The last constraint that needs to hold is that at most one job can be processed on the machine at one time t. It is achieved by the equation (3.24), which controls if a sum of all jobs in process at time t is ≤ 1 on given machine. For the illustration how the constraint works, Figure 3.3 is attached. It shows that the sum of running jobs at time t on given machine m is affected by a job j only if one of the variable $s_{j,k,m}$ equals to 1 for some $k \in \{max(0, t - p_j + 1), \ldots, t\}$.

The objective of the formulation is given by the equation (3.22), which sums all prices $price_{j,t}$ for consuming the energy by processing the jobs. The prices of consuming an energy are chosen by setting the $s_{j,t,m}$ variable values to 1. The overall consumption cost is minimized.

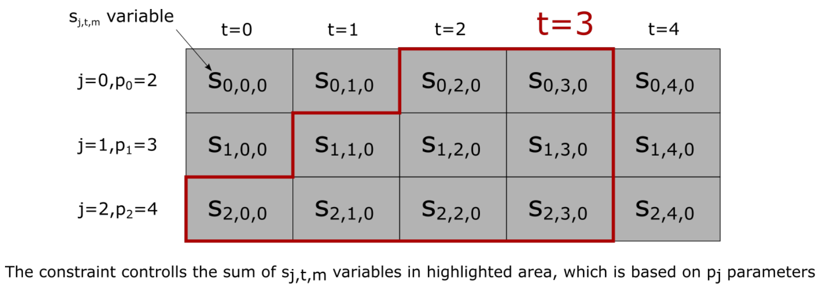


Figure 3.3: Constraint (3.27) controlling, if the sum of jobs in process at time $t = 3$ is ≤ 1

3.3 Two-phase method

In this section, the third method is presented. It is based on decomposition the time-indexed formulation from Section 3.2 into two phases. The first phase defines start times of all jobs and ensures that at every time point at most m (number of machines) jobs are processed. Based on that, the machines are allocated in the second phase. After the describing the first phase formulation, the polynomial algorithm for an allocation of the machines is presented. Experiments in Chapter 5 show that the time needed for finding the solution is reduced dramatically with the help of the described decomposition. The key property of formulation necessary for an efficient performing of the second phase is described in section 3.3.2.1.

3.3.1 First phase formulation

The price $price_{j,t}$ of consuming an energy for every job $j \in J$ starting at time $t \in T_j$ can be computed in advance in the same manner as in Section 3.2.1.

3.3.1.1 Mathematical model

Sets:

	J	Set of jobs $j \in J$
	T	Set of uniform points in time $t \in T$
	T_j	Set of uniform points in time, in which job j can start
Parameters:		
	m	Number of machines
	$price_{j,t}$	Price of consuming an energy by a processing job j starting at time t
	c_j	Electricity consumption of job j
	p_j	Processing time of job j
Variables:		

$$s_{j,t} = \begin{cases} 1, \text{if job } j \text{ starts at time } t \\ 0, \text{else} \end{cases}$$

Objective

$$z^{SP} = \min \sum_{j \in J} \sum_{t \in T} price_{j,t} \cdot s_{j,t} \tag{3.25}$$

subject to

$$\sum_{t \in T_j} s_{j,t} = 1 \qquad \forall j \in J \tag{3.26}$$

$$\sum_{j \in J} \sum_{k=max(0,t-p_j+1)}^{t} s_{j,t} \leq m \qquad \forall t \in T \tag{3.27}$$

The constraints and the objective are very similar as in the section 3.2.2 with only a two differences. First of the differences is that the dimension of the machines is omitted due to a performing the allocation of the machines in the second phase. The last and important thing is that the equation (3.27) ensures that at most m jobs can be processed at one time point. This constraint is crucial for the second phase.

3.3.2 Second phase

In this subsection, the second phase of the two-phase method is described. It is based on a polynomial algorithm, which allocates the machines for the jobs, whose start times are fixed by the first phase. The start time of job j from the first phase is defined as s_j.

3.3.2.1 Algorithm

When the start time s_j for every job is fixed by the first phase, we can assign the jobs to the machines in the *earliest start time first* order. Whenever we are assigning job j to some unoccupied machine m, all other jobs with earliest start time are already assigned. Then, it holds that no other job is in process on machine m at time points $t \in \{s_j, \ldots, s_j + p_j\}$ and whole job j can be processed on machine m. In other words, processing of job j can not be limited by some other job on given machine m, and there is always some unoccupied machine, which can be allocated for job j due to the equation (3.27) from the first phase formulation. Therefore, the algorithm finds a feasible machine assignments.

Without loss of generality we renumber jobs in set J so they are sorted according to s_j, i.e. $s_1 \leq \ldots \leq s_{|J|}$. After that, the simple and correct algorithm for an allocation of the machines is defined as follows:

Algorithm 1: Algorithm for an allocation identical parallel machines

```
renumber jobs s_1 ≤ ... ≤ s_|J|;
for j = 1...|J| do
    m = 1;
    while !isOccupied(m, s_j) do
        m = m + 1;
    end
    allocate machine m for job j;
end
```

Procedure *isOccupied*(m, s_j) returns if machine m is occupied by some other job j' at time s_j, i.e if $s_j \in \{s_{j'}, \ldots, s_{j'} + p_{j'}\}$ for some $j' \in J, j \neq j'$ in case that machine m is allocated for j'.

It is important to sort the jobs according to their start times s_j, because there could be a problem with assigning the jobs to the machines in random order or by indexes of the jobs. The problem that could happen is illustrated in Figure 3.4 together with the feasible allocation performed by the algorithm presented in this section. The goal is to assign a set of jobs $j \in \{0, \ldots, 3\}$ to machines $m \in \{0, 1\}$. Start times s_j of the jobs are defined by the first phase of the method. When the jobs are assigned by index j, the job $j = 3$ can not be assigned to some machine without splitting it into two parts, which are both processed on a different machine. Note that the schedule from the first phase is feasible because the constraint (3.31) mentioned before holds.

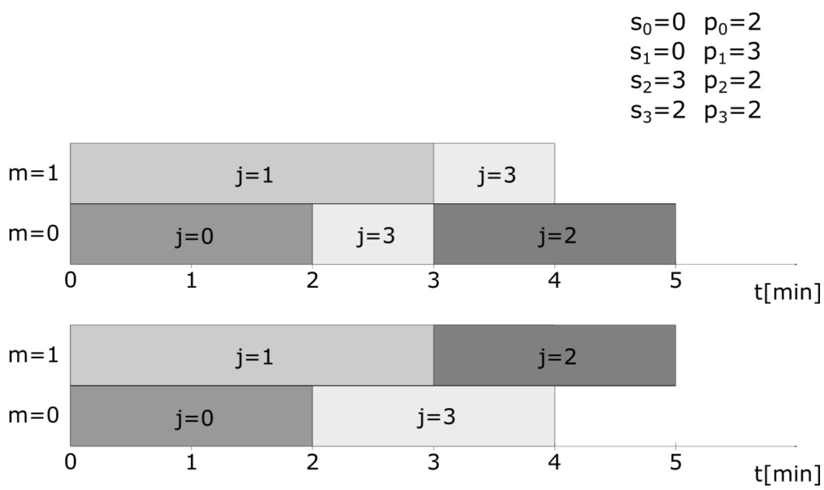


Figure 3.4: Allocation of the machines

3.4 Logic-Based Benders Decomposition

In this section, the method based on dividing the whole problem into two smaller problems is described. The method is inspired by the technique called Logic-Based Benders Decomposition presented in [14] and briefly described in next subsection. After that, both smaller problems incurred by the decomposition and all steps needed for searching the optimal solution are described.

3.4.1 Logic-Based Benders Decomposition description

The Logic-Based Benders Decomposition is based on a decomposition of the whole given problem into two smaller subproblems. The author of [14] mentions that this approach is often used in practice. For instance, central managers assign tasks to facilities and operation managers develop detailed schedules for tasks assigned to each facility. The work of central managers can be denoted as a master problem and the work of the operation managers as a subproblem. Since the schedule for one or more facilities may prove infeasible, the schedulers can ask the managers for a different allocation of tasks. The asking for a different allocation of tasks corresponds to adding constraints to the master problem that forbid the infeasible assignment. These constraints are called Benders cuts, and the cuts must be devised for each problem class because there is no standard scheme for generating them. This is the difference from a classical Benders Decomposition, where the scheme for generating the cuts is defined. In [14], the cuts are developed for three different objective functions for a facility allocation problem. The classical Benders Decomposition requires the subproblem to be a continuous linear or non-linear problem, and because of that, it is inappropriate for the problem addressed in this work.

The main idea of Logic-Based Benders Decomposition is to fix the primary variable x in the master problem to some values and solve the subproblem using these values. The solution of the subproblem is used for obtaining the Benders cut, which is then used as a valid bound on the optimal value, when the primary variables take other values, or for forbidding the current values of the primary variable because there is no feasible solution found by the subproblem using these values. The relaxations of the original problem can also be developed and included in the master problem formulation to increase the efficiency of the whole method. The method iterates between master problem and subproblem until whole solution tree is gone through. For an illustration, how the method works, the Figure 3.5 follows:

3.4.2 Decomposition of IPMSMEC problem

The IPMSMEC problem was decomposed into master problem and subproblem as follows. The master problem deals with an assigning a *majority* of processing time p_j of every job $j \in J$ to some time period n, which is then a *majority interval* for job j, where the majority of the processing time of job j have to be processed. The majority is defined as follows: $majority_j = min(p_j/2, D)$. It states that the majority represents at least D or half of the processing time of job j, which have to be processed in the majority interval n. A set N_j of

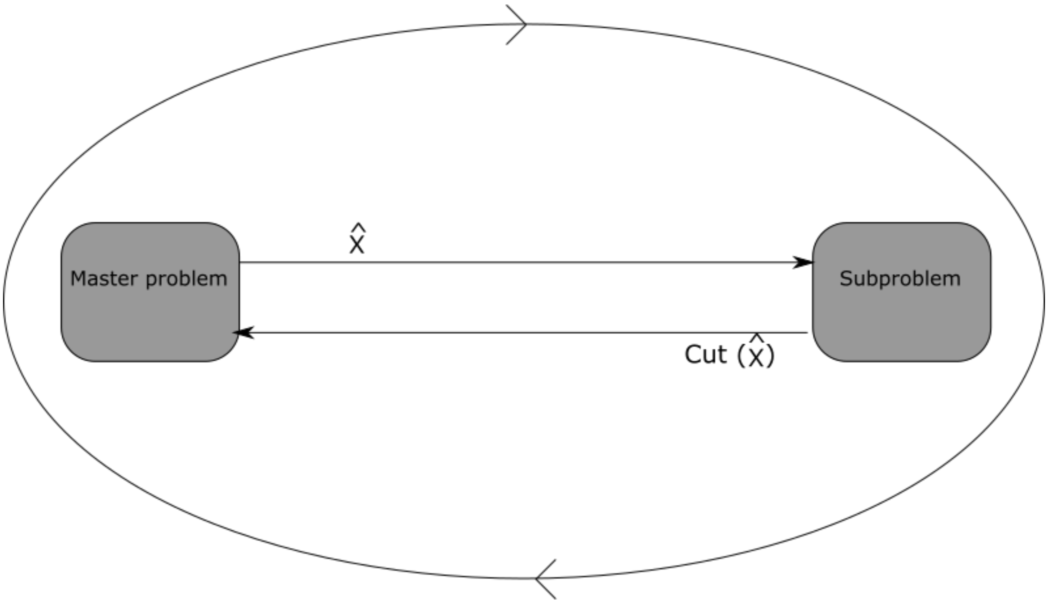


Figure 3.5: Logic-Based Benders Decomposition

possible majority intervals can be defined for every job j with respect to the release time r_j and the deadline d_j as follows:

$$N_j = \{(r_j + majority_j)/D, \ldots, (d_j - majority_j)/D\}$$

The binary variable deciding the assignment of the majority $majority_j$ of the processing time p_j to the period n is $x_{j,n}$.

When interval n is the majority interval for job j (i.e. $x_{j,n} = 1$), the time window T_j can be computed, which represents a set of possible start times of job j so that the $majority_j$ is still processed in period n. T_j is computed by following formula:

$$T_j = \{max(r_j, n \cdot D - (p_j - majority_j)), \ldots, min((n+1) \cdot D - majority_j, d_j - p_j)\}$$

The time-window T_j based on the majority interval n for job j is illustrated by the Figure 3.6.

The subproblem tries to find the start time for every job based on time window T_j computed using the value of x_j, n variable. In case that the feasible schedule is found, the Benders cut defining the bound on the optimal value is added to the master problem. In case that no feasible schedule exists for the fixed values of $x_{j,n}$ from the master problem, a different Benders cut is added to the master problem ensuring that the master problem chooses other values of $x_{j,n}$. The alternation between the master and the subproblem iterates until the optimal solution is found (the iterations are denoted as h).

3.4.2.1 Master problem

The assignment of jobs to majority intervals from the master problem can not be random. The master problem needs some parameter related to the price of the energy which can differ

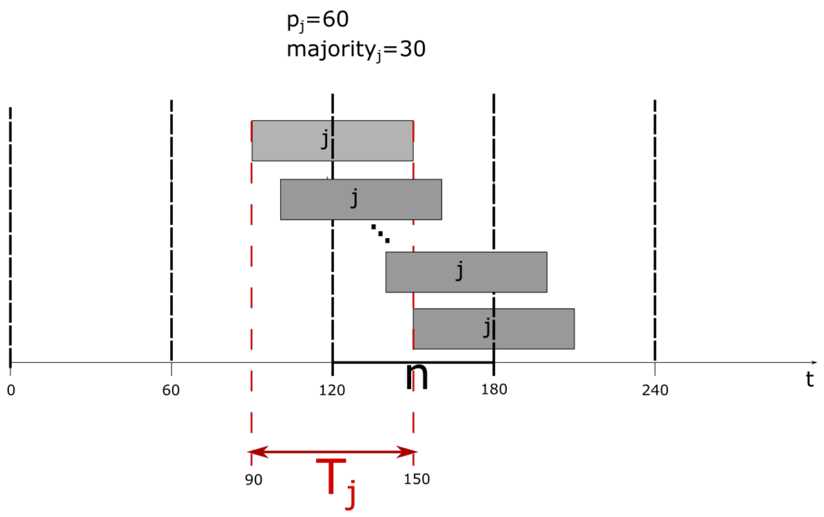


Figure 3.6: Set of possible start times of job based on its majority interval

for every time period n. The parameter is $minPrice_{j,n}$, which stands for the minimal price for the assigning at least the majority $majority_j$ of the processing time p_j to the majority interval n. The $minPrice_{j,n}$ parameter is computed as follows.

For every job $j \in J$ and every $n \in N_j$, the set of possible start times T_j can be defined in case that n is set as the majority interval for job j (see Section 3.4.2). After that, $minPrice_{j,n}$ is defined by following formula:

$$min_{\forall t \in T_j} \sum_{k=t}^{t+p_j} \left(\frac{c_j \cdot price_{\lfloor k/D \rfloor}}{D}\right)$$

Informally, it is the minimal price for processing the job j, so that the majority of its processing time is processed in its majority interval. The parameter takes into account the job energy consumption c_j and the price of the energy $price_n$ in period n. The master problem always tries to find such assignment of the majorities that the overall cost of the consumed energy is as minimal as possible. The value of the objective function defined by the master problem is always the lower bound of an optimal solution since we are using minimal prices instead of actual cost of a job. The MILP model of the master problem is defined as follows:

Sets:

J	Set of jobs $\quad j \in J$
J_n^h	Set of jobs, whose majority was assigned to period n in iteration h
N	Set of uniform time periods $\quad n \in N$
N_j	Set of uniform time periods, to which $majority_j$ of job j can be assigned

Parameters:

c	Energy consumption of all machines
m	Number of machines
p_j	Processing time of job j
r_j	Release-time of job j
d_j	Deadline of job j
D	Time periods length
H	Following iteration of solving the problem
h	Iteration of solving the problem; $h \in 1, ..., H-1$
$minPrice_{j,n}$	Minimal price for the assigning at least majority $majority_j$ to the period n
z_h^{SP}	Value of the subproblem solution obtained in h-th iteration of subproblem
P	Big integer

Variables:

$x_{j,n}$	$= 1$, if the majority $majority_j$ of job j is assigned to period n
	$= 0$ else

Objective

$$\min z^{MP} \tag{3.28}$$

subject to

$$z^{MP} \geq \sum_{n \in N} \sum_{j \in J} x_{j,n} \cdot minPrice_{j,n} \tag{3.29}$$

$$m \cdot D \geq \sum_{j \in J} x_{j,n} \cdot majority_j \qquad \forall n \in N \tag{3.30}$$

$$\sum_{n \in N_j} x_{j,n} = 1 \qquad \forall j \in J \tag{3.31}$$

$$P \cdot \sum_{n \in N} \sum_{j \in J_n^h} (1 - x_{j,n}) + z^{MP} \geq z_h^{SP} \qquad \forall h \in 1, \ldots, H-1 \tag{3.32}$$

$$\sum_{n \in N} \sum_{j \in J_n^h} x_{j,n} \leq |J| - 1 \qquad \forall h \in 1, \ldots, H-1 \tag{3.33}$$

The goal is to minimize the objective z^{MP}, which is always lower bounded by the sum of the minimal prices for processing the majorities of the jobs in their majority intervals (inequality (3.29)). The constraint (3.30) ensures that a capacity of all machines together will not be exceeded by a sum of the majorities of all jobs assigned to the given period. The equation (3.31) tells that the majority of every job is assigned to exactly one period from the set of possible periods. The two remaining inequalities stand for the cuts added from the subproblem. The cut (3.32) defines the lower bound of the value of the objective for some fixed assignment of the variables $x_{j,n}$. It means that the objective z^{MP} of the master problem is greater or equal to the minimal feasible objective z_h^{SP} value obtained in subproblem in h-th iteration of solving the problem for some fixed values of the variables

$x_{j,n}$. When the values are changed, the classical Big-M technique is used for fulfilling the inequality. The cut (3.33) ensures that some fixed assignment of the variables $x_{j,n}$, which causes an infeasibility of the schedule produced by subproblem, will be cut. The meaning is that at least for one job, the majority interval is chosen from a set of periods, which were not chosen as a majority intervals for some job in h-th iteration of solving the master problem.

3.4.2.2 Subproblem

As mentioned before, the subproblem tries to find the best possible schedule for some fixed values of the variables $x_{j,n}$. Firstly, the previously described time-window T_j of possible start times is counted (see Section 3.4.2). The subproblem is basically method 3.3. The difference is that the possible start times for every job are limited to set T_j resulting in much higher effectiveness in solving the subproblem. Note that the allocation of the machines is done by the algorithm presented in 3.3.2.

Chapter 4

Design of heuristic methods

In this chapter, two heuristic methods for solving the IPMSMEC problem are presented:

- Changing a granularity of time for Two-phase method
- Changing release time/deadline time-windows

Note that we wanted the heuristic methods to be based on the exact ones because we want to exploit the effectiveness of the Two-phase method.

4.1 Changing a granularity of time

The first heuristic method is based on changing the granularity of time. It means that the scheduling horizon is divided into longer time periods than one minute long. The length of time periods is defined by input parameter *granularity*. It means that the release times are shifted to the first nearest subsequent time point r'_j, which is multiple of *granularity*, given by following formula:

$$r'_j = \lceil \frac{r_j}{granularity} \rceil \cdot granularity$$

The deadlines are shifted to the first nearest precedent time point d'_j, which is multiple of *granularity*, given by following formula:

$$d'_j = \lfloor \frac{d_j}{granularity} \rfloor \cdot granularity$$

The Two-phase method described in 3.3 is then used for solving the instances of the problem only with the difference that the start times have to be multiples of *granularity*. Note that the number of $s_{j,t}$ variables is reduced, so the solution space is reduced too and the time needed for finding the solution is shorter than in the original method. The changing of the *granularity* causes that the solution doesn't have to be equal to the optimal solution found by the Two-phase method because of the reduction of a number of $s_{j,t}$ variables and a shortening of release times/deadlines time-windows. More precisely, the objective of solution found by this heuristic will never be less than the optimal solution obtained by the exact methods.

4.2 Changing release time/deadline time-windows

The second heuristic method is based on changing the release time/deadline time-windows of jobs. Firstly, the jobs $j \in J$ are sorted according to value $(d_j - r_j)/p_j$ in descending order. This value represents how much the job covers its time window defined by the release time and deadline. The bigger the value is, the less is the possibility to shift the job. After that, we can find the period for every job according to the sorted sequence, which will be the *majority interval* of job. The majority interval n for job j is chosen according to parameter $minprice_{j,n}$ as it is done by master problem in Logic Based Benders Decomposition method (see Section 3.4.2). Every time period has a capacity defined as $m \cdot D$, which can not be exceeded by the sum of $majority_j$ (see Section 3.4.2) of assigned jobs. When the job is assigned to some period n, its release time/deadline time-window is changed according to the new set of possible start times defined as:

$$T_j = \{\max(r_j, n \cdot D - (p_j - majority_j)), \ldots, \min((n+1) \cdot D - majority_j, d_j - p_j)\}$$

After the assignment of the jobs, the Two-phase method 3.3 is started, and some feasible solution is searched. When the assignment to the periods is infeasible, the capacity of r most occupied period is reduced by D, and the new assignment is created. The r parameter is incremented in every iteration.

Algorithm 2: Algorithm for the Changing release time/deadline time-windows method

```
solved = false;
r = 1;
while !solved do
    for j = 1 ... |J| do
        n =findCheapestPeriod(j);
        assignToPeriod(j, n);
        changeRDWindow(j, n);
    end
    solve();
    if feasible() then
        solved = true;
    else
        for i = 0 ... r do
            n =findMostOccupied();
            reduceCapacity(n);
        end
        r++;
        initRDWindows();
    end
end
```

The methods used by an algorithm are defined as follows:

- *findCheapestPeriod(j)* method finds the majority interval n, where the cost of processing of $majority_j$ is minimal according to $minPrice_{j,n}$, and the capacity of the interval will not be exceeded by the $majority_j$.
- *assignToPeriod(j, n)* method reduces the capacity of period n by $majority_j$.
- *changeRDWindow(j, n)* method changes r_j to r'_j and d_j to d'_j according to period n, which is chosen as majority interval for job j. The release time is computed by this formula:
$$r'_j = \max(r_j, n \cdot D - (p_j - majority_j))$$
The deadline is computed by this formula:
$$d'_j = \min((n+1) \cdot D - majority_j + p_j, d_j)$$
- *solve()* method solves the problem by Two-phase method.
- *feasible()* method returns true, if the solution of the problem is feasible.
- *findMostOccupied()* finds the most occupied period by $majority_j$ values of all the jobs $j \in J$.
- *reduceCapacity()* method reduces the capacity of period n by D so that fewer jobs can be assigned to it in next iteration of the algorithm.
- *initRDWindows()* method sets the r_j and d_j of all jobs to initial values.

The consequence of assigning the jobs to the periods in order according to the value $(d_j - r_j)/p_j$ is that the jobs with less possibility to be shifted are fixed to the cheapest possible periods and the other jobs can be assigned to the others, but still according to $minPrice_j, n$ parameter so that the overall electricity consumption cost is minimized.

Chapter 5

Experiments and results

In this chapter, the different approaches for a generating of test instances for the IPMSMEC problem are presented. After that, the results of experiments performed on all designed methods follow. It is followed by a case study for the presentation of usage of the designed methods in practice.

5.1 Test instances generating

Firstly, we need to set a number of the jobs for scheduling ($|J|$) and a number of the identical parallel machines ($|M|$). After that, every instance of addressed problem is denoted by four different types of parameters to generate:

- processing time
- release time/deadline
- price of the energy
- consumption of energy by the jobs

The parameters of every type can be generated by several different methods. Every method of generating one type of parameters can be combined with the other methods, so a relatively large number of different instances may be created.

The first type of parameters is processing time of jobs. The processing times are generated by three different methods:

- PT1: choosing a random integer from the uniform distribution $[0, 120]$ representing durations in minutes
- PT2: randomly selecting between processing time 30,60 or 120 minutes long
- PT3: very similar to the PT1 method, but the distribution is $[0, 60]$

The methods for generating release time/deadline, where the parameters are generated for each job independently, are:

- RD1: randomly picking the release time and the deadline from whole scheduling horizon (the inequality $d_j \geq r_j + p_j$ needs to hold)
- RD2: setting the release time of all jobs to the start of the scheduling horizon, and randomly generating the deadline as in the first method
- RD3: randomly generating the release time, and setting all the deadlines to the end of the scheduling horizon Γ (inequality $r_j + p_j \leq \Gamma$ needs to hold in this case)
- RD4: setting the release time of all jobs to the start of the scheduling horizon, and all the deadlines to the end of the scheduling horizon Γ
- RD5: generating the release time r_j from the distribution $[0, \alpha \sum_{j \in J} p_j]$ and generating the number $d_j - (r_j + p_j)$ from the distribution $[0, \beta \sum_{j \in J} p_j]$ (the parameters are: $\alpha = 0.3, \beta = 0.7$) [3]

The third type of parameters is the price of energy for every hour of the scheduling horizon. Two types of pricing are used for every instance:

- EP1: pricing based on a trading on the day-ahead market
- EP2: pricing based on a trading on the block market

The instance of prices of electricity traded on the day-ahead market was previously used in [19]. The prices from block market were found on OTE, a. s. web pages [16].

The last type of parameters is the consumption of energy for every job $j \in J$. Two types of generating the consumptions were used:

- C1: generating the consumption c_j from the distribution $[1, 10]$
- C2: setting the consumption of all jobs to 1

5.2 Experiments

In this section, the results of the experiments performed on all the algorithms are presented. The experiments were divided into two groups. First, we evaluated the algorithms on small instances, therefore the exact methods can not solve the larger ones in acceptable time. The size of the instance is given by a number of machines and jobs to be scheduled. The exact methods together with the heuristics were used for performing the experiments on small instances. After that, we evaluated the heuristics on bigger instances resulting in the second group of the experiments. The sizes ($|J|$x$|M|$) of the instances of both groups are:

- First type of experiments
  - $|J| = 20, |M| = 4$
  - $|J| = 35, |M| = 7$
  - $|J| = 50, |M| = 10$

  - $|J| = 60, |M| = 12$
- Second type of experiments
  - $|J| = 100, |M| = 8$
  - $|J| = 135, |M| = 10$
  - $|J| = 170, |M| = 12$
  - $|J| = 200, |M| = 15$

As mentioned in section 5.1, the experiments on one size of the instances consists of 60 combinations of the methods of generating the parameters of the instances. For every combination, 10 distinct instances were generated and the results (value of the objective function, gap, and computation time) were averaged. The described approach avoids the results to be distorted in a case that one of the instances is an outlier. It is important to note that the time limit for solving one instance is 60 seconds. After the limit is reached, the computation is interrupted, and the best feasible value of the objective function is presented in the results. It is possible that some method is not able to find a feasible solution in the time limit. In this case, the results are omitted.

The experiments were performed on server with Intel Xeon E5-2620 v2 @ 2.10 GHz processor and 64 GB of RAM. The CPLEX V12.6.3 solver together with Java was used for implementation of all proposed methods.

5.2.1 Small instances

The results of the experiments with small instances performed on the exact methods are presented in the Tables 5.1-5.4 and followed by the Tables 5.5-5.8 presenting the results of performing the experiments on the heuristics. The methods in the tables are noted as follows:

- Continuous formulation (3.1): CONTINUOUS
- Time-indexed formulation (3.2): TIME-INDEXED
- Two-phase method (3.3): TWO-PHASE
- Logic-Based Benders Decomposition (3.4): LBBD
- Changing a granularity of time (4.1): GRANULARITY
- Changing release time/deadline time-window (4.2): R/D WINDOWS

The results of the experiments are always divided into three columns:

- Exact methods
  - Gap[%]: A deviation from the optimal solution
  - t[s]: Computation time with limit of 60 seconds
  - Feas: A number of feasible solutions found from 10 possible in the time limit

- Heuristic methods
  - Gap[%]: A deviation from the optimal solution
  - t[s]: Computation time with limit of 60 seconds
  - Obj: Best feasible solution value

The *Gap* column is computed as $\frac{obj-opt}{opt}$, where *obj* is the best value of feasible solution found and *opt* is optimal value of solution. Note that the *obj* is always greater or equal to *opt*.

As mentioned before, the deviation from the optimal solution, computation time and the best feasible solution value is the average of the results after solving 10 distinct instances. The combinations of methods of generating the parameters of the instances are specified in *Parameters* column. It is also important to note that the granularity of the first heuristic method (GRANULARITY) were set to 15.

5.2.1.1 Analysis of the results

It is first necessary to point at the fact that for the bigger instances, some methods are not even able to find a feasible solution in the time limit. It is caused by the size of the model, which has to be built (a large number of variables etc.), and the limit is already exceeded by it, or by the fact that no branch, which leads to the feasible solution, is found in the limit because of the size of the model. The described problem applies especially to the CONTINUOUS formulation and the TIME-INDEXED formulation, which is documented in Table 5.4. It is interesting that it happens on different types of instances for both methods. The CONTINUOUS formulation is not able to handle the cases when the release times and the deadlines windows are set to the whole scheduling horizon (RD4), which is given by the large number of possible start times. Compared to that, the TIME-INDEXED formulation is not able to handle the cases, when the release times and the deadlines are generated by the methods RD2 and RD5.

For the smallest instances, the TIME-INDEXED formulation did not exceed the time limit so often as the CONTINUOUS formulation, but for the bigger instances, the TIME-INDEXED formulation proved to be even slower and not able to find feasible solutions in the time limit. It is caused by the fact, that the number of variables grows faster with the size of the instance for the TIME-INDEXED formulation.

As illustrated by the Table 5.1, which presents the results for the smallest instances, solving the instances with the price of electricity based on day-ahead market (EP1) especially by the CONTINUOUS formulation and the LBBD method takes much more computation time than solving the instances with prices based on block market(EP2). This observation does not apply so much to the TIME-INDEXED and TWO-PHASE methods. For the last mentioned methods holds that the computation time is changing very similarly, but with the difference that the time consumed by the TWO-PHASE method is a fraction of the time consumed by the TIME-INDEXED formulation. It is the proof that the dividing the TIME-INDEXED formulation into two phases is a huge performance improvement. Another fact is that solving the instances with constant electricity consumption of the jobs (C2) is more complex than with different consumptions (C1). It is given by a greater number of possible combinations of scheduled jobs.

Sometimes it happened that the method exceeded the time limit, but a deviation from the optimal solution (gap) is 0%. It is observable e.g. from Table 5.4 from the results for the TIME-INDEXED formulation, and the fact arises in cases when the method already found the optimal solution, but it didn't manage to get through the whole solution tree.

The LBBD method exceeded the time limit frequently. It is given by the fact, that the number of the possible assignments of the *majorities* is very large and all the assignments have to be solved by the subproblem. On the other hand, it always found some assignment, which gives at least feasible solution. It is in contrast with the other methods, which where exceeding the limit too.

From the Table 5.4, which presents the results of the exact methods on the most complex instances, we can see that all the methods except the TWO-PHASE method are not able to solve them in the time limit. The computation time for the TWO-PHASE method even did not come close to the limit, so there are relatively large reserves. The fact that the TWO-PHASE method is the fastest and the most accurate (always found the optimal solution) method of the tested methods, which is also documented by the aggregated result table 5.9, is the most important observation from the results of the experiments on the exact methods.

The heuristics are very effective in contrast with the exact methods, and the GRANULARITY method proved it the most. It is given by the effectiveness of the TWO-PHASE method on which is the GRANULARITY method based. The difference between these two methods is that in the GRANULARITY method, the number of the variables is greatly reduced, which leads to speed improvement. The most interesting column from the Tables 5.5-5.8 is the gap. It shows how big is the deviation between the best solution found by the heuristic and the optimal solution. It is obvious that the GRANULARITY heuristic is more accurate. It is interesting that for the R/D WINDOWS heuristic, the gap is often higher on smaller instances. The other important thing is that the heuristics were always able to find some feasible solution. Sometimes, the solution was even equal to the optimum.

Parameters	CONTINUOUS			TIME-INDEXED			TWO-PHASE			LBBD		
	Gap[%]	t[s]	Feas	Gap[%]	t[s]	Feas	Gap[%]	t[s]	Feas	Gap[%]	t[s]	Feas
PT1RD1EP1C1	0,00	22,26	10/10	0,00	7,06	10/10	0,00	0,75	10/10	0,00	18,29	10/10
PT1RD1EP1C2	0,00	15,77	10/10	0,00	7,12	10/10	0,00	0,77	10/10	0,01	18,12	10/10
PT1RD1EP2C1	0,00	2,82	10/10	0,00	5,86	10/10	0,00	0,61	10/10	0,00	2,34	10/10
PT1RD1EP2C2	0,00	4,84	10/10	0,00	7,39	10/10	0,00	0,82	10/10	0,15	7,42	10/10
PT1RD2EP1C1	0,17	60,00	10/10	0,00	36,08	10/10	0,00	4,39	10/10	10,19	60,00	10/10
PT1RD2EP1C2	0,06	60,00	10/10	5,98	46,58	10/10	0,00	5,48	10/10	6,84	60,00	10/10
PT1RD2EP2C1	0,00	2,94	10/10	0,00	17,28	10/10	0,00	1,57	10/10	0,00	4,37	10/10
PT1RD2EP2C2	0,00	2,87	10/10	0,00	17,22	10/10	0,00	1,58	10/10	1,50	10,68	10/10
PT1RD3EP1C1	1,23	60,00	10/10	0,00	28,58	10/10	0,00	2,71	10/10	16,45	60,00	10/10
PT1RD3EP1C2	0,06	60,00	10/10	0,00	32,45	10/10	0,00	8,93	10/10	6,30	60,00	10/10
PT1RD3EP2C1	0,00	9,42	10/10	0,00	16,70	10/10	0,00	1,99	10/10	0,37	32,16	10/10
PT1RD3EP2C2	0,00	15,65	10/10	0,00	22,18	10/10	0,00	5,29	10/10	1,98	28,03	10/10
PT1RD4EP1C1	0,28	60,00	10/10	0,00	39,33	10/10	0,00	4,61	10/10	7,81	60,00	10/10
PT1RD4EP1C2	0,22	60,00	10/10	0,00	50,91	10/10	0,00	6,97	10/10	12,28	60,00	10/10
PT1RD4EP2C1	0,00	3,70	10/10	0,00	28,50	10/10	0,00	3,23	10/10	5,39	31,62	10/10
PT1RD4EP2C2	0,00	3,82	10/10	0,00	24,46	10/10	0,00	2,97	10/10	1,35	10,67	10/10
PT1RD5EP1C1	0,10	60,00	10/10	0,00	11,88	10/10	0,00	1,27	10/10	0,77	60,00	10/10
PT1RD5EP1C2	0,37	60,00	10/10	0,00	20,89	10/10	0,00	1,81	10/10	0,63	60,00	10/10
PT1RD5EP2C1	0,00	8,60	10/10	0,00	9,41	10/10	0,00	1,01	10/10	0,02	28,50	10/10
PT1RD5EP2C2	0,00	7,31	10/10	0,00	11,23	10/10	0,00	1,06	10/10	0,60	25,45	10/10
PT2RD1EP1C1	0,00	32,29	10/10	0,00	8,82	10/10	0,00	0,93	10/10	0,02	24,79	10/10
PT2RD1EP1C2	0,00	28,56	10/10	0,00	8,29	10/10	0,00	0,85	10/10	0,31	25,71	10/10
PT2RD1EP2C1	0,00	9,24	10/10	0,00	8,14	10/10	0,00	0,78	10/10	0,00	9,71	10/10
PT2RD1EP2C2	0,00	16,66	10/10	0,00	9,72	10/10	0,00	1,05	10/10	0,41	18,37	10/10
PT2RD2EP1C1	0,39	60,00	10/10	0,00	32,40	10/10	0,00	3,16	10/10	8,05	60,00	10/10
PT2RD2EP1C2	0,08	60,00	10/10	0,00	37,04	10/10	0,00	3,85	10/10	18,11	60,00	10/10
PT2RD2EP2C1	0,00	2,71	10/10	0,00	21,88	10/10	0,00	2,03	10/10	2,25	20,25	10/10
PT2RD2EP2C2	0,00	2,95	10/10	0,00	21,08	10/10	0,00	1,98	10/10	2,31	16,13	10/10
PT2RD3EP1C1	1,93	60,00	10/10	0,00	23,00	10/10	0,00	2,32	10/10	24,95	60,00	10/10
PT2RD3EP1C2	0,21	60,00	10/10	0,00	19,60	10/10	0,00	2,11	10/10	21,71	60,00	10/10
PT2RD3EP2C1	0,00	27,74	10/10	0,00	16,64	10/10	0,00	1,90	10/10	4,68	33,95	10/10
PT2RD3EP2C2	0,00	39,97	10/10	0,00	19,02	10/10	0,00	1,99	10/10	1,56	48,15	10/10
PT2RD4EP1C1	0,23	60,00	10/10	0,00	40,38	10/10	0,00	4,58	10/10	11,13	60,00	10/10
PT2RD4EP1C2	0,04	60,00	10/10	0,68	43,73	10/10	0,00	4,53	10/10	33,00	60,00	10/10
PT2RD4EP2C1	0,00	3,12	10/10	0,00	29,72	10/10	0,00	3,38	10/10	0,00	20,05	10/10
PT2RD4EP2C2	0,00	3,73	10/10	0,00	30,25	10/10	0,00	3,38	10/10	2,86	24,93	10/10
PT2RD5EP1C1	0,34	60,00	10/10	0,00	15,25	10/10	0,00	1,50	10/10	4,20	60,00	10/10
PT2RD5EP1C2	0,10	60,00	10/10	0,00	16,32	10/10	0,00	1,62	10/10	0,65	60,00	10/10
PT2RD5EP2C1	0,00	26,89	10/10	0,00	14,73	10/10	0,00	1,38	10/10	0,34	29,38	10/10
PT2RD5EP2C2	0,00	4,82	10/10	0,00	13,31	10/10	0,00	1,28	10/10	0,20	20,90	10/10
PT3RD1EP1C1	0,00	9,27	10/10	0,00	3,11	10/10	0,00	0,43	10/10	0,00	0,49	10/10
PT3RD1EP1C2	0,00	3,68	10/10	0,00	2,91	10/10	0,00	0,41	10/10	0,00	1,08	10/10
PT3RD1EP2C1	0,00	1,43	10/10	0,00	2,72	10/10	0,00	0,41	10/10	0,00	0,12	10/10
PT3RD1EP2C2	0,00	1,64	10/10	0,00	2,67	10/10	0,00	0,42	10/10	0,00	0,11	10/10
PT3RD2EP1C1	0,06	60,00	10/10	0,00	15,34	10/10	0,00	1,94	10/10	0,24	60,00	10/10
PT3RD2EP1C2	0,01	60,00	10/10	0,00	16,68	10/10	0,00	2,37	10/10	13,72	60,00	10/10
PT3RD2EP2C1	0,00	1,58	10/10	0,00	6,19	10/10	0,00	0,83	10/10	0,00	0,22	10/10
PT3RD2EP2C2	0,00	1,59	10/10	0,00	5,43	10/10	0,00	0,79	10/10	0,00	0,66	10/10
PT3RD3EP1C1	0,34	60,00	10/10	0,00	10,30	10/10	0,00	1,16	10/10	2,51	60,00	10/10
PT3RD3EP1C2	0,03	60,00	10/10	0,00	11,28	10/10	0,00	1,59	10/10	2,88	60,00	10/10
PT3RD3EP2C1	0,00	2,56	10/10	0,00	6,54	10/10	0,00	0,90	10/10	0,00	4,20	10/10
PT3RD3EP2C2	0,00	2,35	10/10	0,00	5,90	10/10	0,00	0,83	10/10	0,00	1,51	10/10
PT3RD4EP1C1	0,04	60,20	10/10	0,00	19,42	10/10	0,00	2,28	10/10	9,93	60,00	10/10
PT3RD4EP1C2	0,03	60,20	10/10	0,00	24,95	10/10	0,00	3,55	10/10	9,84	60,00	10/10
PT3RD4EP2C1	0,00	4,52	10/10	0,00	10,30	10/10	0,00	1,57	10/10	0,00	3,05	10/10
PT3RD4EP2C2	0,00	5,02	10/10	0,00	9,89	10/10	0,00	1,53	10/10	0,00	1,80	10/10
PT3RD5EP1C1	0,04	60,00	10/10	0,00	4,47	10/10	0,00	0,62	10/10	0,21	60,00	10/10
PT3RD5EP1C2	0,01	60,00	10/10	0,00	6,23	10/10	0,00	0,84	10/10	0,15	60,00	10/10
PT3RD5EP2C1	0,00	0,90	10/10	0,00	1,65	10/10	0,00	0,25	10/10	0,00	0,10	10/10
PT3RD5EP2C2	0,00	1,18	10/10	0,00	2,29	10/10	0,00	0,32	10/10	0,00	0,11	10/10

Table 5.1: The results of the exact methods on the instances with size 20x4

Parameters	CONTINUOUS			TIME-INDEXED			TWO-PHASE			LBBD		
	Gap[%]	t[s]	Feas	Gap[%]	t[s]	Feas	Gap[%]	t[s]	Feas	Gap[%]	t[s]	Feas
PT1RD1EP1C1	0,14	60,00	10/10	0,00	23,71	10/10	0,00	1,50	10/10	0,01	26,08	10/10
PT1RD1EP1C2	0,07	60,00	10/10	0,00	21,54	10/10	0,00	1,46	10/10	0,02	12,16	10/10
PT1RD1EP2C1	0,00	37,96	10/10	0,00	22,69	10/10	0,00	1,39	10/10	0,00	1,98	10/10
PT1RD1EP2C2	0,00	32,14	10/10	0,00	24,69	10/10	0,00	1,43	10/10	0,00	8,36	10/10
PT1RD2EP1C1	1,50	60,00	10/10	29,03	60,00	10/10	0,00	8,79	10/10	16,74	60,00	10/10
PT1RD2EP1C2	0,30	60,00	10/10	30,11	60,00	10/10	0,00	14,40	10/10	23,89	60,00	10/10
PT1RD2EP2C1	0,00	14,01	10/10	2,83	59,63	10/10	0,00	3,17	10/10	0,00	17,70	10/10
PT1RD2EP2C2	0,00	14,30	10/10	4,35	60,00	10/10	0,00	3,44	10/10	0,00	23,58	10/10
PT1RD3EP1C1	7,98	60,00	10/10	25,42	60,00	10/10	0,00	4,81	10/10	23,83	60,00	10/10
PT1RD3EP1C2	4,87	60,00	10/10	19,67	60,00	10/10	0,00	7,86	10/10	12,11	60,00	10/10
PT1RD3EP2C1	0,12	42,54	10/10	1,48	52,83	10/10	0,00	6,08	10/10	3,10	41,61	10/10
PT1RD3EP2C2	0,17	37,04	10/10	2,09	51,75	10/10	0,00	4,60	10/10	2,89	35,82	10/10
PT1RD4EP1C1	2,08	60,00	10/10	2,88	60,00	10/10	0,00	8,76	10/10	25,13	60,00	10/10
PT1RD4EP1C2	0,88	60,00	10/10	2,35	60,00	10/10	0,00	12,53	10/10	23,91	60,00	10/10
PT1RD4EP2C1	0,00	28,41	10/10	0,00	60,00	10/10	0,00	5,33	10/10	0,00	13,82	10/10
PT1RD4EP2C2	0,00	24,06	10/10	0,00	60,00	10/10	0,00	5,42	10/10	1,30	11,13	10/10
PT1RD5EP1C1	4,37	60,00	10/10	6,73	55,97	10/10	0,00	3,56	10/10	2,93	60,00	10/10
PT1RD5EP1C2	5,97	60,00	10/10	16,10	59,69	10/10	0,00	4,91	10/10	4,68	60,00	10/10
PT1RD5EP2C1	0,00	16,82	10/10	0,00	29,27	10/10	0,00	1,56	10/10	0,02	16,87	10/10
PT1RD5EP2C2	0,00	22,77	10/10	0,00	32,86	10/10	0,00	1,64	10/10	0,19	19,87	10/10
PT2RD1EP1C1	0,25	60,00	10/10	0,00	27,41	10/10	0,00	1,75	10/10	0,00	24,16	10/10
PT2RD1EP1C2	0,16	60,00	10/10	0,00	28,09	10/10	0,00	1,77	10/10	0,34	20,99	10/10
PT2RD1EP2C1	0,00	45,04	10/10	0,00	29,12	10/10	0,00	1,69	10/10	0,00	12,78	10/10
PT2RD1EP2C2	0,03	54,27	10/10	0,00	28,63	10/10	0,00	1,57	10/10	0,02	19,91	10/10
PT2RD2EP1C1	3,37	60,00	10/10	-	-	9/10	0,00	6,35	10/10	28,73	60,00	10/10
PT2RD2EP1C2	5,87	60,00	10/10	41,38	60,00	10/10	0,00	7,59	10/10	41,67	60,00	10/10
PT2RD2EP2C1	0,00	20,89	10/10	9,52	60,00	10/10	0,00	3,56	10/10	0,04	27,39	10/10
PT2RD2EP2C2	0,00	17,55	10/10	6,71	60,00	10/10	0,00	3,39	10/10	0,11	27,11	10/10
PT2RD3EP1C1	20,12	60,00	10/10	15,93	60,00	10/10	0,00	4,10	10/10	19,60	60,00	10/10
PT2RD3EP1C2	12,95	60,00	10/10	12,67	60,00	10/10	0,00	3,76	10/10	28,37	60,00	10/10
PT2RD3EP2C1	0,39	42,50	10/10	4,27	60,00	10/10	0,00	3,72	10/10	6,92	52,84	10/10
PT2RD3EP2C2	0,47	47,97	10/10	2,45	58,23	10/10	0,00	4,18	10/10	5,66	46,83	10/10
PT2RD4EP1C1	5,61	60,00	10/10	4,42	60,00	10/10	0,00	9,14	10/10	34,62	60,00	10/10
PT2RD4EP1C2	10,16	60,00	10/10	2,95	60,00	10/10	0,00	9,57	10/10	38,79	60,00	10/10
PT2RD4EP2C1	0,00	21,99	10/10	0,00	60,00	10/10	0,00	5,84	10/10	7,86	60,00	10/10
PT2RD4EP2C2	0,00	30,51	10/10	0,03	60,00	10/10	0,00	6,15	10/10	2,70	18,73	10/10
PT2RD5EP1C1	11,21	60,00	10/10	10,82	55,50	10/10	0,00	3,49	10/10	9,53	60,00	10/10
PT2RD5EP1C2	12,45	60,00	10/10	5,15	57,03	10/10	0,00	3,37	10/10	17,85	60,00	10/10
PT2RD5EP2C1	0,27	43,50	10/10	2,62	51,44	10/10	0,00	2,84	10/10	0,33	39,15	10/10
PT2RD5EP2C2	0,27	46,11	10/10	2,50	53,23	10/10	0,00	2,87	10/10	4,25	45,99	10/10
PT3RD1EP1C1	0,00	27,83	10/10	0,00	8,42	10/10	0,00	0,67	10/10	0,00	0,16	10/10
PT3RD1EP1C2	0,00	40,13	10/10	0,00	9,73	10/10	0,00	0,76	10/10	0,00	0,16	10/10
PT3RD1EP2C1	0,00	10,16	10/10	0,00	9,01	10/10	0,00	0,68	10/10	0,00	0,16	10/10
PT3RD1EP2C2	0,00	7,26	10/10	0,00	9,18	10/10	0,00	0,72	10/10	0,00	0,26	10/10
PT3RD2EP1C1	0,22	60,00	10/10	2,50	47,22	10/10	0,00	3,62	10/10	4,92	60,00	10/10
PT3RD2EP1C2	0,18	60,00	10/10	15,61	60,00	10/10	0,00	4,36	10/10	0,21	60,00	10/10
PT3RD2EP2C1	0,00	5,25	10/10	0,00	19,92	10/10	0,00	1,55	10/10	0,00	0,52	10/10
PT3RD2EP2C2	0,00	4,73	10/10	0,00	21,74	10/10	0,00	1,53	10/10	0,00	0,21	10/10
PT3RD3EP1C1	2,51	60,00	10/10	1,99	42,51	10/10	0,00	2,57	10/10	2,55	60,00	10/10
PT3RD3EP1C2	1,18	60,00	10/10	0,00	40,08	10/10	0,00	2,88	10/10	4,61	60,00	10/10
PT3RD3EP2C1	0,00	11,14	10/10	0,00	19,00	10/10	0,00	1,71	10/10	0,00	18,75	10/10
PT3RD3EP2C2	0,00	10,99	10/10	0,00	19,30	10/10	0,00	1,56	10/10	0,00	1,81	10/10
PT3RD4EP1C1	0,73	60,00	10/10	2,75	60,00	10/10	0,00	5,25	10/10	0,28	60,00	10/10
PT3RD4EP1C2	0,36	60,00	10/10	3,49	60,00	10/10	0,00	6,58	10/10	12,73	60,00	10/10
PT3RD4EP2C1	0,00	24,31	10/10	0,00	36,33	10/10	0,00	3,01	10/10	0,00	0,79	10/10
PT3RD4EP2C2	0,00	25,22	10/10	0,00	33,54	10/10	0,00	2,86	10/10	0,00	0,17	10/10
PT3RD5EP1C1	0,19	60,00	10/10	0,00	21,14	10/10	0,00	1,29	10/10	4,02	60,00	10/10
PT3RD5EP1C2	0,07	60,00	10/10	0,26	40,32	10/10	0,00	1,90	10/10	0,62	60,00	10/10
PT3RD5EP2C1	0,00	8,00	10/10	0,00	6,87	10/10	0,00	0,48	10/10	0,00	0,18	10/10
PT3RD5EP2C2	0,00	4,25	10/10	0,00	5,73	10/10	0,00	0,42	10/10	0,00	0,18	10/10

Table 5.2: The results of the exact methods on the instances with size 35x7

Parameters	CONTINUOUS			TIME-INDEXED			TWO-PHASE			LBBD		
	Gap[%]	t[s]	Feas	Gap[%]	t[s]	Feas	Gap[%]	t[s]	Feas	Gap[%]	t[s]	Feas
PT1RD1EP1C1	2,04	60,00	10/10	2,25	53,31	10/10	0,00	2,44	10/10	0,00	12,19	10/10
PT1RD1EP1C2	0,52	60,00	10/10	0,00	50,05	10/10	0,00	2,21	10/10	0,00	18,23	10/10
PT1RD1EP2C1	0,07	57,74	10/10	0,63	46,83	10/10	0,00	2,01	10/10	0,00	6,23	10/10
PT1RD1EP2C2	0,05	60,00	10/10	0,59	50,36	10/10	0,00	2,21	10/10	0,00	0,29	10/10
PT1RD2EP1C1	15,52	60,00	10/10	-	-	2/10	0,00	16,65	10/10	28,72	60,00	10/10
PT1RD2EP1C2	23,10	60,00	10/10	-	-	0/10	0,00	32,86	10/10	43,06	60,00	10/10
PT1RD2EP2C1	1,56	51,52	10/10	-	-	3/10	0,00	4,42	10/10	1,36	34,89	10/10
PT1RD2EP2C2	0,24	50,38	10/10	-	-	1/10	0,00	4,39	10/10	0,19	37,03	10/10
PT1RD3EP1C1	26,25	60,00	10/10	43,13	60,00	10/10	0,00	7,80	10/10	22,32	60,00	10/10
PT1RD3EP1C2	17,21	60,00	10/10	39,78	60,00	10/10	0,00	18,65	10/10	17,56	60,00	10/10
PT1RD3EP2C1	6,13	60,00	10/10	13,79	60,00	10/10	0,00	6,50	10/10	2,38	49,74	10/10
PT1RD3EP2C2	7,70	60,00	10/10	14,58	60,00	10/10	0,00	17,79	10/10	3,91	46,68	10/10
PT1RD4EP1C1	16,99	60,00	10/10	3,06	60,00	10/10	0,00	13,73	10/10	13,92	60,00	10/10
PT1RD4EP1C2	6,75	60,00	10/10	2,30	60,00	10/10	0,00	23,25	10/10	41,82	60,00	10/10
PT1RD4EP2C1	-	-	0/10	0,00	60,00	10/10	0,00	7,32	10/10	2,67	24,00	10/10
PT1RD4EP2C2	-	-	9/10	0,00	60,00	10/10	0,00	7,91	10/10	0,00	12,09	10/10
PT1RD5EP1C1	42,07	60,00	10/10	23,99	60,00	10/10	0,00	7,87	10/10	16,81	60,00	10/10
PT1RD5EP1C2	42,54	60,00	10/10	30,28	60,00	10/10	0,00	8,37	10/10	7,46	60,00	10/10
PT1RD5EP2C1	1,76	56,37	10/10	7,26	60,00	10/10	0,00	2,80	10/10	0,12	32,55	10/10
PT1RD5EP2C2	1,75	54,18	10/10	7,16	59,52	10/10	0,00	2,37	10/10	0,29	21,94	10/10
PT2RD1EP1C1	2,29	60,00	10/10	3,87	54,78	10/10	0,00	2,48	10/10	0,00	6,26	10/10
PT2RD1EP1C2	2,37	60,00	10/10	7,41	59,88	10/10	0,00	2,64	10/10	0,00	6,24	10/10
PT2RD1EP2C1	0,08	60,00	10/10	1,06	52,70	10/10	0,00	2,25	10/10	0,00	0,60	10/10
PT2RD1EP2C2	0,02	60,00	10/10	2,23	60,00	10/10	0,00	2,62	10/10	0,00	10,63	10/10
PT2RD2EP1C1	20,79	60,00	10/10	-	-	0/10	0,00	9,04	10/10	46,80	60,00	10/10
PT2RD2EP1C2	17,56	60,00	10/10	-	-	1/10	0,00	10,97	10/10	42,44	60,00	10/10
PT2RD2EP2C1	0,24	50,07	10/10	-	-	1/10	0,00	4,83	10/10	0,06	35,66	10/10
PT2RD2EP2C2	1,80	60,00	10/10	-	-	0/10	0,00	5,02	10/10	0,71	47,71	10/10
PT2RD3EP1C1	30,92	60,00	10/10	39,53	60,00	10/10	0,00	5,58	10/10	34,13	60,00	10/10
PT2RD3EP1C2	25,49	60,00	10/10	33,57	60,00	10/10	0,00	5,44	10/10	28,10	60,00	10/10
PT2RD3EP2C1	10,18	60,00	10/10	14,96	60,00	10/10	0,00	6,16	10/10	7,95	54,97	10/10
PT2RD3EP2C2	8,67	60,00	10/10	13,33	60,00	10/10	0,00	6,16	10/10	5,63	48,23	10/10
PT2RD4EP1C1	10,44	60,00	10/10	3,59	60,00	10/10	0,00	13,27	10/10	41,65	60,00	10/10
PT2RD4EP1C2	9,82	60,00	10/10	2,72	60,00	10/10	0,00	13,56	10/10	39,68	60,00	10/10
PT2RD4EP2C1	1,22	54,66	10/10	0,00	60,00	10/10	0,00	8,64	10/10	6,45	60,00	10/10
PT2RD4EP2C2	1,16	58,02	10/10	0,00	60,00	10/10	0,00	8,97	10/10	2,10	16,70	10/10
PT2RD5EP1C1	45,54	60,00	10/10	-	-	6/10	0,00	5,29	10/10	16,35	60,00	10/10
PT2RD5EP1C2	46,51	60,00	10/10	-	-	6/10	0,00	7,01	10/10	16,49	60,00	10/10
PT2RD5EP2C1	6,53	60,00	10/10	-	-	8/10	0,00	3,74	10/10	8,82	49,23	10/10
PT2RD5EP2C2	2,72	60,00	10/10	-	-	6/10	0,00	3,72	10/10	3,74	33,91	10/10
PT3RD1EP1C1	0,01	60,00	10/10	0,00	20,58	10/10	0,00	1,16	10/10	0,00	0,20	10/10
PT3RD1EP1C2	0,00	60,00	10/10	0,00	18,73	10/10	0,00	1,08	10/10	0,00	0,21	10/10
PT3RD1EP2C1	0,00	40,57	10/10	0,00	18,75	10/10	0,00	1,04	10/10	0,00	0,22	10/10
PT3RD1EP2C2	0,00	41,99	10/10	0,00	21,33	10/10	0,00	1,18	10/10	0,00	0,24	10/10
PT3RD2EP1C1	0,86	60,00	10/10	34,08	60,00	10/10	0,00	6,17	10/10	5,23	60,00	10/10
PT3RD2EP1C2	7,63	60,00	10/10	32,02	60,00	10/10	0,00	8,27	10/10	23,39	60,00	10/10
PT3RD2EP2C1	0,00	17,84	10/10	1,49	48,30	10/10	0,00	2,41	10/10	0,00	1,48	10/10
PT3RD2EP2C2	0,00	17,16	10/10	0,00	42,17	10/10	0,00	2,19	10/10	0,00	0,26	10/10
PT3RD3EP1C1	11,86	60,00	10/10	14,85	60,00	10/10	0,00	3,49	10/10	2,74	60,00	10/10
PT3RD3EP1C2	10,73	60,00	10/10	22,84	60,00	10/10	0,00	4,80	10/10	6,69	60,00	10/10
PT3RD3EP2C1	0,00	45,63	10/10	0,00	40,40	10/10	0,00	2,33	10/10	0,00	3,46	10/10
PT3RD3EP2C2	0,49	52,19	10/10	0,00	37,36	10/10	0,00	2,30	10/10	0,00	0,34	10/10
PT3RD4EP1C1	-	-	8/10	5,61	60,00	10/10	0,00	8,31	10/10	27,14	60,00	10/10
PT3RD4EP1C2	-	-	9/10	4,89	60,00	10/10	0,00	10,15	10/10	41,55	60,00	10/10
PT3RD4EP2C1	-	-	0/10	0,00	60,00	10/10	0,00	4,06	10/10	0,00	0,27	10/10
PT3RD4EP2C2	-	-	8/10	0,00	60,00	10/10	0,00	4,27	10/10	0,00	0,21	10/10
PT3RD5EP1C1	13,71	60,00	10/10	1,13	54,56	10/10	0,00	2,07	10/10	4,18	60,00	10/10
PT3RD5EP1C2	5,89	60,00	10/10	4,11	59,34	10/10	0,00	2,84	10/10	0,65	60,00	10/10
PT3RD5EP2C1	0,00	26,21	10/10	0,00	13,00	10/10	0,00	0,62	10/10	0,00	0,21	10/10
PT3RD5EP2C2	0,00	28,38	10/10	0,00	13,66	10/10	0,00	0,63	10/10	0,00	0,29	10/10

Table 5.3: The results of the exact methods on the instances with size 50x10

Parameters	CONTINUOUS			TIME-INDEXED			TWO-PHASE			LBBD		
	Gap[%]	t[s]	Feas	Gap[%]	t[s]	Feas	Gap[%]	t[s]	Feas	Gap[%]	t[s]	Feas
PT1RD1EP1C1	6,39	60,00	10/10	14,87	60,00	10/10	0,00	2,64	10/10	0,00	6,30	10/10
PT1RD1EP1C2	9,07	60,00	10/10	12,07	60,00	10/10	0,00	2,75	10/10	0,00	0,28	10/10
PT1RD1EP2C1	1,89	60,00	10/10	4,99	60,00	10/10	0,00	2,74	10/10	0,00	0,66	10/10
PT1RD1EP2C2	2,58	60,00	10/10	4,27	60,00	10/10	0,00	2,66	10/10	0,00	0,75	10/10
PT1RD2EP1C1	29,21	60,00	10/10	-	-	0/10	0,00	18,07	10/10	33,15	60,00	10/10
PT1RD2EP1C2	33,86	60,00	10/10	-	-	0/10	0,00	39,31	10/10	44,42	60,00	10/10
PT1RD2EP2C1	6,17	59,31	10/10	-	-	1/10	0,00	5,80	10/10	1,53	27,91	10/10
PT1RD2EP2C2	4,61	60,00	10/10	-	-	0/10	0,00	5,66	10/10	1,95	53,78	10/10
PT1RD3EP1C1	30,60	60,00	10/10	43,93	60,00	10/10	0,00	10,83	10/10	25,90	60,00	10/10
PT1RD3EP1C2	25,37	60,00	10/10	39,13	60,00	10/10	0,00	14,25	10/10	21,85	60,00	10/10
PT1RD3EP2C1	6,38	60,00	10/10	14,50	60,00	10/10	0,00	9,23	10/10	6,62	54,39	10/10
PT1RD3EP2C2	7,31	60,00	10/10	14,23	60,00	10/10	0,00	20,14	10/10	8,07	55,65	10/10
PT1RD4EP1C1	41,37	60,00	10/10	2,66	60,00	10/10	0,00	15,81	10/10	34,49	60,00	10/10
PT1RD4EP1C2	-	-	9/10	2,12	60,00	10/10	0,00	29,51	10/10	45,54	60,00	10/10
PT1RD4EP2C1	1,37	60,00	10/10	0,00	60,00	10/10	0,00	9,10	10/10	2,94	38,53	10/10
PT1RD4EP2C2	-	-	6/10	0,00	60,00	10/10	0,00	9,39	10/10	0,00	12,97	10/10
PT1RD5EP1C1	29,40	60,00	10/10	-	-	5/10	0,00	9,15	10/10	15,97	60,00	10/10
PT1RD5EP1C2	33,04	60,00	10/10	-	-	9/10	0,00	13,62	10/10	20,15	60,00	10/10
PT1RD5EP2C1	4,64	60,00	10/10	-	-	6/10	0,00	3,13	10/10	3,11	48,65	10/10
PT1RD5EP2C2	6,06	60,00	10/10	-	-	8/10	0,00	2,91	10/10	0,12	24,82	10/10
PT2RD1EP1C1	6,91	60,00	10/10	20,88	60,00	10/10	0,00	3,11	10/10	0,00	6,32	10/10
PT2RD1EP1C2	8,13	60,00	10/10	21,95	60,00	10/10	0,00	3,39	10/10	0,81	12,27	10/10
PT2RD1EP2C1	5,24	60,00	10/10	6,05	60,00	10/10	0,00	3,01	10/10	0,00	1,37	10/10
PT2RD1EP2C2	3,02	60,00	10/10	6,08	60,00	10/10	0,00	3,18	10/10	0,01	6,61	10/10
PT2RD2EP1C1	20,55	60,00	10/10	-	-	0/10	0,00	14,84	10/10	46,68	60,00	10/10
PT2RD2EP1C2	18,71	60,00	10/10	-	-	0/10	0,00	12,88	10/10	47,73	60,00	10/10
PT2RD2EP2C1	2,66	59,71	10/10	-	-	0/10	0,00	6,24	10/10	0,59	41,70	10/10
PT2RD2EP2C2	4,02	60,00	10/10	-	-	0/10	0,00	6,05	10/10	0,94	59,18	10/10
PT2RD3EP1C1	32,63	60,00	10/10	42,56	60,00	10/10	0,00	6,47	10/10	35,65	60,00	10/10
PT2RD3EP1C2	26,31	60,00	10/10	36,60	60,00	10/10	0,00	6,08	10/10	30,94	60,00	10/10
PT2RD3EP2C1	9,73	60,00	10/10	14,10	60,00	10/10	0,00	8,16	10/10	13,48	60,00	10/10
PT2RD3EP2C2	9,31	60,00	10/10	13,64	60,00	10/10	0,00	6,53	10/10	5,55	60,00	10/10
PT2RD4EP1C1	43,33	60,00	10/10	3,99	60,00	10/10	0,00	14,12	10/10	42,92	60,00	10/10
PT2RD4EP1C2	42,27	60,00	10/10	2,49	60,00	10/10	0,00	14,49	10/10	44,92	60,00	10/10
PT2RD4EP2C1	10,59	60,00	10/10	0,00	60,00	10/10	0,00	10,58	10/10	9,22	56,37	10/10
PT2RD4EP2C2	10,98	60,00	10/10	0,00	60,00	10/10	0,00	10,60	10/10	0,00	1,01	10/10
PT2RD5EP1C1	41,62	60,00	10/10	-	-	2/10	0,00	7,87	10/10	36,67	60,00	10/10
PT2RD5EP1C2	39,50	60,00	10/10	-	-	4/10	0,00	8,29	10/10	45,27	60,00	10/10
PT2RD5EP2C1	10,88	60,00	10/10	-	-	0/10	0,00	4,71	10/10	3,53	56,34	10/10
PT2RD5EP2C2	9,69	60,00	10/10	-	-	1/10	0,00	4,59	10/10	2,43	43,81	10/10
PT3RD1EP1C1	0,27	60,00	10/10	0,00	28,05	10/10	0,00	1,22	10/10	0,00	0,25	10/10
PT3RD1EP1C2	1,15	60,00	10/10	0,00	28,15	10/10	0,00	1,30	10/10	0,00	0,24	10/10
PT3RD1EP2C1	0,00	58,83	10/10	0,00	27,77	10/10	0,00	1,23	10/10	0,00	0,24	10/10
PT3RD1EP2C2	0,02	59,25	10/10	0,00	28,64	10/10	0,00	1,29	10/10	0,00	0,29	10/10
PT3RD2EP1C1	13,86	60,00	10/10	34,19	60,00	10/10	0,00	7,44	10/10	36,82	60,00	10/10
PT3RD2EP1C2	7,49	60,00	10/10	38,50	60,00	10/10	0,00	11,02	10/10	34,55	60,00	10/10
PT3RD2EP2C1	0,78	33,78	10/10	6,27	60,00	10/10	0,00	2,92	10/10	0,00	0,33	10/10
PT3RD2EP2C2	0,20	34,06	10/10	8,49	60,00	10/10	0,00	3,25	10/10	0,00	0,33	10/10
PT3RD3EP1C1	13,31	60,00	10/10	29,63	60,00	10/10	0,00	4,57	10/10	8,72	60,00	10/10
PT3RD3EP1C2	15,57	60,00	10/10	25,34	60,00	10/10	0,00	6,04	10/10	9,22	60,00	10/10
PT3RD3EP2C1	1,56	60,00	10/10	1,69	56,40	10/10	0,00	2,83	10/10	0,12	29,07	10/10
PT3RD3EP2C2	1,23	60,00	10/10	0,98	57,86	10/10	0,00	2,87	10/10	0,00	5,29	10/10
PT3RD4EP1C1	-	-	0/10	5,81	60,00	10/10	0,00	10,87	10/10	41,13	60,00	10/10
PT3RD4EP1C2	-	-	0/10	5,23	60,00	10/10	0,00	10,58	10/10	45,56	60,00	10/10
PT3RD4EP2C1	-	-	0/10	0,00	60,00	10/10	0,00	5,01	10/10	0,00	0,40	10/10
PT3RD4EP2C2	-	-	0/10	0,00	60,00	10/10	0,00	4,96	10/10	0,00	0,25	10/10
PT3RD5EP1C1	11,52	60,00	10/10	1,78	60,00	10/10	0,00	2,22	10/10	2,61	60,00	10/10
PT3RD5EP1C2	16,34	60,00	10/10	2,82	60,00	10/10	0,00	3,24	10/10	4,78	60,00	10/10
PT3RD5EP2C1	0,09	31,67	10/10	0,00	18,61	10/10	0,00	0,69	10/10	0,00	0,25	10/10
PT3RD5EP2C2	0,00	26,87	10/10	0,00	15,13	10/10	0,00	0,60	10/10	0,00	0,22	10/10

Table 5.4: The results of the exact methods on the instances with size 60x12

Parameters	GRANULARITY			R/D WINDOWS		
	Gap[%]	t[s]	Obj	Gap[%]	t[s]	Obj
PT1RD1EP1C1	1,66	0,04	4613,50	0,12	0,08	4543,81
PT1RD1EP1C2	1,91	0,03	868,96	0,37	0,12	856,36
PT1RD1EP2C1	0,44	0,03	4330,85	0,53	0,11	4337,41
PT1RD1EP2C2	0,51	0,03	759,87	0,56	0,15	760,73
PT1RD2EP1C1	0,38	0,11	3171,94	17,58	0,53	4001,21
PT1RD2EP1C2	0,40	0,13	549,26	12,87	0,64	658,61
PT1RD2EP2C1	0,00	0,05	3664,56	3,87	0,74	3827,46
PT1RD2EP2C2	0,00	0,05	647,90	0,99	0,47	655,12
PT1RD3EP1C1	3,54	0,07	3592,68	11,11	0,32	3919,61
PT1RD3EP1C2	3,33	0,13	611,94	4,50	0,33	624,74
PT1RD3EP2C1	0,13	0,05	3542,80	2,66	0,27	3651,51
PT1RD3EP2C2	0,49	0,09	683,69	2,04	0,31	695,95
PT1RD4EP1C1	0,21	0,10	2994,86	3,17	2,18	3113,66
PT1RD4EP1C2	0,31	0,08	533,35	0,34	1,65	533,46
PT1RD4EP2C1	0,00	0,05	4016,31	1,32	3,09	4059,07
PT1RD4EP2C2	0,00	0,05	648,23	0,69	1,40	652,91
PT1RD5EP1C1	1,82	0,06	3070,24	7,41	0,37	3331,76
PT1RD5EP1C2	3,07	0,10	630,34	9,77	0,33	707,55
PT1RD5EP2C1	0,07	0,04	3579,12	1,22	0,25	3624,67
PT1RD5EP2C2	0,08	0,04	680,59	0,13	0,16	680,87
PT2RD1EP1C1	1,29	0,03	5596,57	0,12	0,11	5533,67
PT2RD1EP1C2	1,62	0,03	1036,09	0,63	0,13	1027,09
PT2RD1EP2C1	0,32	0,03	5121,02	0,44	0,16	5127,60
PT2RD1EP2C2	0,47	0,03	937,81	0,62	0,27	939,57
PT2RD2EP1C1	0,03	0,08	3454,43	10,55	0,75	3896,35
PT2RD2EP1C2	0,04	0,06	646,30	12,43	0,78	745,80
PT2RD2EP2C1	0,00	0,05	4274,50	5,78	1,27	4528,80
PT2RD2EP2C2	0,00	0,05	796,80	2,05	0,83	812,64
PT2RD3EP1C1	0,27	0,07	4057,90	16,55	0,45	4901,41
PT2RD3EP1C2	0,19	0,08	882,74	7,35	0,42	954,16
PT2RD3EP2C1	0,04	0,05	3923,40	3,49	0,40	4075,59
PT2RD3EP2C2	0,06	0,06	733,68	2,77	0,42	754,72
PT2RD4EP1C1	0,00	0,10	3167,56	10,85	1,18	3666,03
PT2RD4EP1C2	0,00	0,07	641,62	16,62	1,05	821,68
PT2RD4EP2C1	0,00	0,05	4118,46	9,68	2,61	4589,91
PT2RD4EP2C2	0,00	0,05	783,52	4,24	2,44	820,14
PT2RD5EP1C1	1,61	0,05	4269,62	11,48	0,34	4844,57
PT2RD5EP1C2	1,98	0,05	762,45	8,56	0,31	835,33
PT2RD5EP2C1	0,22	0,04	4361,20	0,69	0,22	4383,78
PT2RD5EP2C2	0,09	0,04	725,22	0,03	0,20	724,65
PT3RD1EP1C1	1,50	0,04	2330,19	0,01	0,06	2297,24
PT3RD1EP1C2	1,35	0,03	389,65	0,00	0,07	384,10
PT3RD1EP2C1	0,13	0,03	1939,71	0,01	0,08	1937,44
PT3RD1EP2C2	0,35	0,03	341,56	0,00	0,08	340,42
PT3RD2EP1C1	0,18	0,05	1511,77	0,32	0,39	1513,86
PT3RD2EP1C2	0,16	0,07	267,99	0,13	0,39	267,92
PT3RD2EP2C1	0,00	0,05	1711,68	0,00	0,39	1711,68
PT3RD2EP2C2	0,00	0,05	325,08	0,00	0,52	325,08
PT3RD3EP1C1	1,70	0,09	1552,03	5,13	0,26	1613,53
PT3RD3EP1C2	1,71	0,07	303,77	4,95	0,27	316,66
PT3RD3EP2C1	0,00	0,05	1921,34	0,53	0,16	1932,60
PT3RD3EP2C2	0,00	0,05	329,45	0,84	0,14	332,38
PT3RD4EP1C1	0,08	0,07	1434,78	1,16	0,48	1454,94
PT3RD4EP1C2	0,04	0,06	263,33	1,19	0,49	267,01
PT3RD4EP2C1	0,00	0,04	1795,95	3,08	0,71	1874,47
PT3RD4EP2C2	0,00	0,05	320,32	1,47	0,68	326,45
PT3RD5EP1C1	0,20	0,04	1466,32	0,46	0,33	1470,24
PT3RD5EP1C2	0,22	0,04	260,07	0,09	0,31	259,72
PT3RD5EP2C1	0,00	0,03	1719,54	0,00	0,10	1719,54
PT3RD5EP2C2	0,00	0,03	355,79	0,00	0,12	355,79

Table 5.5: The results of the heuristic methods on the instances with size 20x4

Parameters	GRANULARITY			R/D WINDOWS		
	Gap[%]	t[s]	Obj	Gap[%]	t[s]	Obj
PT1RD1EP1C1	1,58	0,04	8675,02	0,12	0,12	8543,03
PT1RD1EP1C2	1,34	0,04	1506,36	0,02	0,13	1486,57
PT1RD1EP2C1	0,51	0,04	6953,14	0,24	0,22	6934,77
PT1RD1EP2C2	0,43	0,04	1246,92	0,63	0,27	1249,86
PT1RD2EP1C1	0,38	0,13	4963,99	3,09	1,65	5129,01
PT1RD2EP1C2	0,29	0,10	979,96	0,56	1,43	982,81
PT1RD2EP2C1	0,00	0,06	6155,06	0,00	0,80	6155,06
PT1RD2EP2C2	0,00	0,07	1204,72	0,57	0,95	1213,19
PT1RD3EP1C1	2,90	0,15	6013,65	8,24	0,69	6382,74
PT1RD3EP1C2	2,85	0,11	1179,47	0,99	0,61	1159,49
PT1RD3EP2C1	0,12	0,09	6627,79	1,56	0,50	6722,53
PT1RD3EP2C2	0,30	0,14	1166,68	2,10	0,49	1187,88
PT1RD4EP1C1	0,27	0,11	5827,23	13,77	0,97	7359,62
PT1RD4EP1C2	0,22	0,10	884,45	3,07	0,82	911,88
PT1RD4EP2C1	0,00	0,07	6245,47	0,00	1,63	6245,47
PT1RD4EP2C2	0,00	0,07	1175,39	0,00	0,84	1175,39
PT1RD5EP1C1	1,56	0,13	5418,20	3,65	0,77	5543,54
PT1RD5EP1C2	2,52	0,07	1039,47	7,91	0,78	1111,88
PT1RD5EP2C1	0,00	0,05	6388,12	0,00	0,28	6388,12
PT1RD5EP2C2	0,00	0,05	1132,29	0,44	0,33	1138,19
PT2RD1EP1C1	1,33	0,04	9654,03	0,02	0,13	9528,14
PT2RD1EP1C2	1,55	0,04	1845,42	0,25	0,20	1820,51
PT2RD1EP2C1	0,27	0,04	8126,17	0,08	0,18	8109,70
PT2RD1EP2C2	0,39	0,04	1513,88	0,58	0,27	1516,53
PT2RD2EP1C1	0,02	0,13	6050,99	7,16	2,00	6593,83
PT2RD2EP1C2	0,09	0,10	1188,16	13,05	2,20	1380,31
PT2RD2EP2C1	0,00	0,07	7913,22	5,50	1,86	8420,68
PT2RD2EP2C2	0,00	0,06	1339,62	2,38	1,51	1375,83
PT2RD3EP1C1	0,29	0,07	6611,46	7,11	0,68	7115,12
PT2RD3EP1C2	0,22	0,10	1462,44	1,71	0,80	1485,40
PT2RD3EP2C1	0,00	0,07	7812,75	2,02	1,00	7983,27
PT2RD3EP2C2	0,07	0,08	1351,80	2,17	0,87	1382,71
PT2RD4EP1C1	0,00	0,12	6141,49	13,46	1,16	7290,66
PT2RD4EP1C2	0,00	0,10	1115,92	16,51	1,21	1402,24
PT2RD4EP2C1	0,00	0,07	7596,16	1,78	3,28	7793,83
PT2RD4EP2C2	0,00	0,07	1402,70	0,38	1,37	1408,15
PT2RD5EP1C1	1,33	0,06	6404,61	13,24	0,96	7364,13
PT2RD5EP1C2	1,27	0,07	1222,07	11,29	0,82	1361,57
PT2RD5EP2C1	0,08	0,05	7298,68	0,12	0,39	7301,49
PT2RD5EP2C2	0,05	0,06	1369,30	0,32	0,52	1373,11
PT3RD1EP1C1	1,15	0,04	3780,15	0,00	0,09	3734,98
PT3RD1EP1C2	1,65	0,04	685,51	0,00	0,08	674,19
PT3RD1EP2C1	0,40	0,04	3267,96	0,00	0,11	3254,76
PT3RD1EP2C2	0,31	0,03	620,86	0,00	0,11	618,84
PT3RD2EP1C1	0,21	0,09	2275,85	0,23	0,91	2276,44
PT3RD2EP1C2	0,18	0,08	466,25	0,10	1,05	465,86
PT3RD2EP2C1	0,00	0,06	3322,82	0,00	0,53	3322,82
PT3RD2EP2C2	0,00	0,06	556,10	0,00	0,46	556,10
PT3RD3EP1C1	1,55	0,09	2704,72	1,96	0,50	2715,75
PT3RD3EP1C2	1,39	0,15	485,16	0,19	0,44	479,31
PT3RD3EP2C1	0,00	0,06	3217,36	0,51	0,26	3233,81
PT3RD3EP2C2	0,00	0,06	585,65	0,98	0,26	591,90
PT3RD4EP1C1	0,07	0,08	2546,99	0,47	1,19	2557,18
PT3RD4EP1C2	0,03	0,09	479,87	0,42	1,05	481,76
PT3RD4EP2C1	0,00	0,06	3270,75	0,00	0,56	3270,75
PT3RD4EP2C2	0,00	0,07	573,64	0,00	0,45	573,64
PT3RD5EP1C1	0,23	0,07	2703,77	0,28	0,89	2704,84
PT3RD5EP1C2	0,19	0,06	484,08	0,04	0,88	483,35
PT3RD5EP2C1	0,00	0,03	3373,56	0,00	0,19	3373,56
PT3RD5EP2C2	0,00	0,03	556,76	0,00	0,15	556,76

Table 5.6: The results of the heuristic methods on the instances with size 35x7

Parameters	GRANULARITY			R/D WINDOWS		
	Gap[%]	t[s]	Obj	Gap[%]	t[s]	Obj
PT1RD1EP1C1	1,44	0,06	11606,01	0,00	0,15	11441,87
PT1RD1EP1C2	1,44	0,05	2195,06	0,00	0,17	2163,81
PT1RD1EP2C1	0,36	0,05	9226,05	0,29	0,28	9221,91
PT1RD1EP2C2	0,38	0,05	1898,50	0,12	0,21	1893,77
PT1RD2EP1C1	0,34	0,14	7851,84	1,16	2,90	7919,02
PT1RD2EP1C2	0,31	0,12	1418,07	0,73	2,98	1424,60
PT1RD2EP2C1	0,00	0,07	9262,52	0,00	1,31	9262,52
PT1RD2EP2C2	0,00	0,07	1609,59	0,00	1,37	1609,59
PT1RD3EP1C1	3,36	0,14	9182,54	6,53	1,03	9545,27
PT1RD3EP1C2	3,40	0,11	1718,58	2,76	1,07	1708,67
PT1RD3EP2C1	0,24	0,13	9723,85	2,43	1,20	9952,67
PT1RD3EP2C2	0,71	0,16	1770,70	2,31	1,20	1800,14
PT1RD4EP1C1	0,22	0,16	7429,44	0,82	1,13	7475,90
PT1RD4EP1C2	0,28	0,14	1383,54	0,56	1,28	1387,66
PT1RD4EP2C1	0,00	0,09	8784,89	0,00	1,20	8784,89
PT1RD4EP2C2	0,00	0,09	1707,64	0,00	1,27	1707,64
PT1RD5EP1C1	1,24	0,09	7971,65	4,18	1,72	8280,45
PT1RD5EP1C2	2,04	0,12	1500,45	4,67	1,56	1543,79
PT1RD5EP2C1	0,00	0,06	9506,54	0,00	0,66	9506,68
PT1RD5EP2C2	0,00	0,05	1645,45	0,00	0,47	1645,45
PT2RD1EP1C1	1,18	0,05	12650,20	0,00	0,17	12500,00
PT2RD1EP1C2	1,54	0,05	2455,19	0,04	0,16	2418,43
PT2RD1EP2C1	0,29	0,05	11156,81	0,14	0,23	11140,67
PT2RD1EP2C2	0,34	0,05	2164,97	0,00	0,32	2157,91
PT2RD2EP1C1	0,03	0,15	8839,39	4,47	3,59	9286,33
PT2RD2EP1C2	0,03	0,10	1596,80	1,23	3,48	1616,97
PT2RD2EP2C1	0,00	0,07	9867,04	0,00	1,97	9867,04
PT2RD2EP2C2	0,00	0,07	1960,46	0,00	2,12	1960,46
PT2RD3EP1C1	0,24	0,10	10272,14	6,96	1,10	11027,99
PT2RD3EP1C2	0,33	0,08	2157,49	1,47	1,16	2183,02
PT2RD3EP2C1	0,05	0,09	11407,24	2,36	1,66	11679,32
PT2RD3EP2C2	0,08	0,11	1976,17	1,54	1,56	2007,44
PT2RD4EP1C1	0,00	0,15	8236,78	1,50	1,40	8365,53
PT2RD4EP1C2	0,00	0,13	1571,34	1,81	1,55	1603,52
PT2RD4EP2C1	0,00	0,09	10778,38	0,07	1,85	10787,21
PT2RD4EP2C2	0,00	0,09	2018,56	0,00	1,81	2018,56
PT2RD5EP1C1	1,19	0,09	9930,84	11,21	1,40	11284,10
PT2RD5EP1C2	1,31	0,11	1789,16	11,34	1,53	2004,13
PT2RD5EP2C1	0,00	0,06	9862,32	0,03	0,75	9864,66
PT2RD5EP2C2	0,00	0,06	1884,10	0,03	0,61	1884,79
PT3RD1EP1C1	1,51	0,04	5717,48	0,00	0,09	5632,61
PT3RD1EP1C2	1,52	0,04	986,43	0,00	0,09	971,48
PT3RD1EP2C1	0,32	0,04	4789,91	0,00	0,14	4774,36
PT3RD1EP2C2	0,34	0,04	905,48	0,00	0,15	902,38
PT3RD2EP1C1	0,18	0,09	3437,51	0,23	1,47	3439,35
PT3RD2EP1C2	0,16	0,08	684,68	0,11	1,66	684,35
PT3RD2EP2C1	0,00	0,06	4496,00	0,00	0,81	4496,00
PT3RD2EP2C2	0,00	0,06	826,85	0,00	0,85	826,85
PT3RD3EP1C1	1,23	0,09	3612,19	1,85	0,54	3635,11
PT3RD3EP1C2	1,54	0,14	737,52	0,30	0,65	728,33
PT3RD3EP2C1	0,00	0,07	4595,77	0,33	0,50	4611,23
PT3RD3EP2C2	0,00	0,07	789,72	1,22	0,32	799,67
PT3RD4EP1C1	0,06	0,10	3605,34	0,34	1,66	3615,15
PT3RD4EP1C2	0,05	0,11	714,56	0,28	2,21	716,16
PT3RD4EP2C1	0,00	0,08	4357,11	0,00	0,61	4357,11
PT3RD4EP2C2	0,00	0,08	861,21	0,00	0,69	861,21
PT3RD5EP1C1	0,23	0,07	3756,36	0,29	1,37	3758,58
PT3RD5EP1C2	0,20	0,08	664,59	0,04	1,52	663,50
PT3RD5EP2C1	0,00	0,04	4646,62	0,00	0,20	4646,62
PT3RD5EP2C2	0,00	0,03	848,59	0,00	0,23	848,59

Table 5.7: The results of the heuristic methods on the instances with size 50x10

Parameters	GRANULARITY			R/D WINDOWS		
	Gap[%]	t[s]	Obj	Gap[%]	t[s]	Obj
PT1RD1EP1C1	1,56	0,06	14208,02	0,00	0,15	13989,73
PT1RD1EP1C2	1,27	0,06	2551,87	0,00	0,14	2519,36
PT1RD1EP2C1	0,38	0,06	11996,50	0,22	0,22	11975,43
PT1RD1EP2C2	0,32	0,05	2259,97	0,11	0,28	2255,03
PT1RD2EP1C1	0,34	0,12	9009,47	1,05	5,21	9073,48
PT1RD2EP1C2	0,33	0,12	1664,73	0,18	4,47	1662,37
PT1RD2EP2C1	0,00	0,08	10789,67	0,00	1,96	10789,67
PT1RD2EP2C2	0,00	0,08	2001,68	0,00	1,79	2001,68
PT1RD3EP1C1	3,47	0,10	10970,84	7,82	1,37	11517,27
PT1RD3EP1C2	3,38	0,13	1963,77	1,74	1,46	1931,21
PT1RD3EP2C1	0,18	0,14	11325,63	2,18	1,46	11561,03
PT1RD3EP2C2	0,53	0,14	2090,21	2,16	1,53	2124,36
PT1RD4EP1C1	0,22	0,17	9370,66	1,07	1,76	9450,84
PT1RD4EP1C2	0,33	0,16	1657,57	0,80	1,51	1665,68
PT1RD4EP2C1	0,00	0,11	10850,98	0,00	1,78	10850,98
PT1RD4EP2C2	0,00	0,10	2037,26	0,00	1,76	2037,26
PT1RD5EP1C1	1,62	0,13	9500,69	5,35	1,77	9965,99
PT1RD5EP1C2	1,33	0,14	1630,16	2,76	1,94	1658,61
PT1RD5EP2C1	0,00	0,06	10763,11	0,00	1,56	10763,11
PT1RD5EP2C2	0,00	0,06	1972,25	0,00	0,74	1972,25
PT2RD1EP1C1	1,58	0,06	16896,97	0,00	0,19	16631,69
PT2RD1EP1C2	1,47	0,06	2945,55	0,16	0,28	2907,18
PT2RD1EP2C1	0,39	0,05	13648,59	0,13	0,23	13611,36
PT2RD1EP2C2	0,32	0,06	2551,95	0,12	0,30	2547,27
PT2RD2EP1C1	0,03	0,12	10631,85	2,63	4,83	10928,58
PT2RD2EP1C2	0,01	0,10	1989,12	1,06	4,68	2010,65
PT2RD2EP2C1	0,00	0,09	12627,62	0,00	2,71	12627,62
PT2RD2EP2C2	0,00	0,08	2337,28	0,00	2,56	2337,28
PT2RD3EP1C1	0,23	0,10	12010,00	9,26	1,60	13210,48
PT2RD3EP1C2	0,40	0,09	2290,87	2,76	1,61	2349,74
PT2RD3EP2C1	0,03	0,10	12846,55	1,48	2,37	13044,07
PT2RD3EP2C2	0,07	0,12	2345,77	1,12	2,01	2370,59
PT2RD4EP1C1	0,00	0,15	10205,89	2,48	2,02	10478,35
PT2RD4EP1C2	0,00	0,15	1812,53	0,45	1,68	1820,28
PT2RD4EP2C1	0,00	0,11	12657,50	0,00	2,49	12657,50
PT2RD4EP2C2	0,00	0,11	2362,18	0,00	2,57	2362,18
PT2RD5EP1C1	0,74	0,11	10494,35	9,78	2,59	11666,62
PT2RD5EP1C2	1,07	0,10	2064,02	12,64	2,28	2367,61
PT2RD5EP2C1	0,00	0,06	12835,12	0,00	1,45	12835,12
PT2RD5EP2C2	0,00	0,07	2343,92	0,02	1,25	2344,47
PT3RD1EP1C1	1,35	0,05	6561,20	0,00	0,12	6472,16
PT3RD1EP1C2	1,31	0,04	1291,10	0,00	0,10	1274,25
PT3RD1EP2C1	0,41	0,04	5857,67	0,00	0,16	5834,89
PT3RD1EP2C2	0,41	0,05	1067,20	0,00	0,17	1062,94
PT3RD2EP1C1	0,16	0,08	4435,82	0,26	2,08	4440,66
PT3RD2EP1C2	0,17	0,10	798,24	0,07	2,38	797,43
PT3RD2EP2C1	0,00	0,08	5418,18	0,00	1,01	5418,18
PT3RD2EP2C2	0,00	0,07	1072,08	0,00	1,08	1072,08
PT3RD3EP1C1	1,68	0,14	4198,55	1,72	0,88	4200,60
PT3RD3EP1C2	1,62	0,13	869,24	0,26	0,83	857,29
PT3RD3EP2C1	0,00	0,08	5222,64	0,22	0,39	5234,93
PT3RD3EP2C2	0,00	0,07	1019,46	1,15	0,40	1031,01
PT3RD4EP1C1	0,05	0,12	4303,06	0,35	1,92	4316,28
PT3RD4EP1C2	0,02	0,12	807,77	0,44	2,25	811,24
PT3RD4EP2C1	0,00	0,09	5337,01	0,00	0,76	5337,01
PT3RD4EP2C2	0,00	0,09	986,93	0,00	0,82	986,93
PT3RD5EP1C1	0,24	0,06	4424,94	0,29	1,96	4427,39
PT3RD5EP1C2	0,20	0,05	837,68	0,06	1,69	836,57
PT3RD5EP2C1	0,00	0,04	5524,92	0,00	0,31	5524,92
PT3RD5EP2C2	0,00	0,03	958,26	0,00	0,21	958,26

Table 5.8: The results of the heuristic methods on the instances with size 60x12

Instance size	CONTINUOUS			TIME-INDEXED			TWO-PHASE			LBBD		
	Gap[%]	t[s]	Feas	Gap[%]	t[s]	Feas	Gap[%]	t[s]	Feas	Gap[%]	t[s]	Feas
20x4	0,11	29,58	60/60	0,11	17,31	60/60	0,00	2,09	60/60	4,15	32,72	60/60
35x7	1,93	41,35	60/60	4,93	42,94	59/60	0,00	3,85	60/60	6,60	34,38	60/60
50x10	7,75	54,55	54/60	10,04	50,85	48/60	0,00	5,67	60/60	7,98	32,34	60/60
60x12	11,16	56,96	54/60	12,33	54,23	44/60	0,00	6,12	60/60	10,00	30,97	60/60

Table 5.9: The aggregated results of the exact methods on small instances

Instance size	GRANULARITY			R/D WINDOWS		
	Gap[%]	t[s]	Obj	Gap[%]	t[s]	Obj
20x4	0,57	0,06	1900,37	3,76	0,56	1998,68
35x7	0,49	0,07	3302,70	2,52	0,76	3415,91
50x10	0,50	0,08	4683,37	1,26	1,12	4748,38
60x12	0,49	0,09	5625,08	1,24	1,53	5696,15

Table 5.10: The aggregated results of the heuristic methods on small instances

5.2.2 Large instnaces

In this section, the results of the experiments performed only on the heuristics are presented in the Tables 5.11-5.14. The sizes of the instances are intentionally bigger than in the experiments of the first group, so the exact methods are not able to solve them before the time limit. The results are again divided into three columns:

- t[s]: Computation time with limit of 60 seconds
- Obj: Best feasible solution value
- Feas: A number of feasible solutions found from 10 possible in the time limit

The granularity parameter for the GRANULARITY method was set to 15 as in the first group of experiments.

5.2.2.1 Analysis of the results

It is obvious from the Tables 5.11-5.14 that the GRANULARITY heuristic based on the most effective exact TWO-PHASE method is still very effective on the instances that can not be solved by the exact methods.

Since instances are large, the optimal solutions are not known, so we compare the values of the best feasible solutions found by both methods. As for the first type of experiments, the Obj value is often lower for the GRANULARITY method, so it proved to be more accurate than the R/D WINDOWS method.

The computation time increased significantly for the R/D WINDOWS method for bigger instances. The method was not even be able to find some feasible solution value in time limit sometimes as can be seen in the Table 5.14. The time increased for the instances, where the release times and deadlines were generated by the method RD4. This type of instances proved to be the hardest to solve for all the exact methods and heuristics, which is not surprising because of the increased number of variables caused by the sizes of possible time windows, to which the jobs can be scheduled.

Parameters	GRANULARITY			R/D WINDOWS		
	t[s]	Obj	Feas	t[s]	Obj	Feas
PT1RD1EP1C1	0,15	24890,54	10/10	1,94	25805,27	10/10
PT1RD1EP1C2	0,19	4471,71	10/10	2,20	4469,47	10/10
PT1RD1EP2C1	0,13	20635,38	10/10	1,26	20789,20	10/10
PT1RD1EP2C2	0,14	3722,97	10/10	1,61	3726,33	10/10
PT1RD2EP1C1	0,59	20109,76	10/10	4,23	22663,88	10/10
PT1RD2EP1C2	0,49	4136,64	10/10	6,02	4395,31	10/10
PT1RD2EP2C1	0,55	18525,79	10/10	14,65	19270,03	10/10
PT1RD2EP2C2	0,35	3671,43	10/10	9,39	3699,92	10/10
PT1RD3EP1C1	0,44	23228,32	10/10	3,37	23990,20	10/10
PT1RD3EP1C2	0,37	4290,13	10/10	3,28	4280,13	10/10
PT1RD3EP2C1	0,29	20668,78	10/10	3,10	21098,74	10/10
PT1RD3EP2C2	0,20	3691,89	10/10	1,82	3685,11	10/10
PT1RD4EP1C1	0,61	16990,75	10/10	48,10	21692,26	10/10
PT1RD4EP1C2	0,65	3604,36	10/10	46,86	4331,35	10/10
PT1RD4EP2C1	0,53	18492,98	10/10	20,08	19853,83	10/10
PT1RD4EP2C2	0,56	3484,59	10/10	19,34	3674,12	10/10
PT1RD5EP1C1	0,27	20217,42	10/10	5,55	22257,14	10/10
PT1RD5EP1C2	0,32	3970,09	10/10	5,06	4207,88	10/10
PT1RD5EP2C1	0,28	19624,83	10/10	3,68	20018,64	10/10
PT1RD5EP2C2	0,31	3655,77	10/10	2,10	3654,90	10/10
PT2RD1EP1C1	0,13	29560,56	10/10	5,19	31280,38	10/10
PT2RD1EP1C2	0,15	5247,22	10/10	3,64	5383,24	10/10
PT2RD1EP2C1	0,12	24442,02	10/10	2,88	24895,08	10/10
PT2RD1EP2C2	0,11	4416,71	10/10	2,45	4466,75	10/10
PT2RD2EP1C1	0,33	24937,66	10/10	4,35	27907,69	10/10
PT2RD2EP1C2	0,39	5121,35	10/10	4,79	5312,44	10/10
PT2RD2EP2C1	0,27	22681,90	10/10	10,11	23571,49	10/10
PT2RD2EP2C2	0,24	4133,34	10/10	11,20	4156,84	10/10
PT2RD3EP1C1	0,25	26772,65	10/10	3,88	28612,90	10/10
PT2RD3EP1C2	0,23	5408,44	10/10	4,37	5613,40	10/10
PT2RD3EP2C1	0,19	23878,66	10/10	2,74	24603,00	10/10
PT2RD3EP2C2	0,17	4374,39	10/10	2,64	4386,89	10/10
PT2RD4EP1C1	0,46	20639,49	10/10	29,44	27171,29	10/10
PT2RD4EP1C2	0,40	4552,06	10/10	24,34	5144,80	10/10
PT2RD4EP2C1	0,40	21201,55	10/10	19,09	23278,51	10/10
PT2RD4EP2C2	0,32	4045,92	10/10	19,38	4288,01	10/10
PT2RD5EP1C1	0,24	24997,15	10/10	3,70	28871,49	10/10
PT2RD5EP1C2	0,22	4905,92	10/10	3,84	5160,36	10/10
PT2RD5EP2C1	0,24	22422,20	10/10	3,83	23224,37	10/10
PT2RD5EP2C2	0,19	4204,48	10/10	4,95	4254,62	10/10
PT3RD1EP1C1	0,10	11301,10	10/10	0,33	11113,39	10/10
PT3RD1EP1C2	0,09	2014,21	10/10	0,50	1991,07	10/10
PT3RD1EP2C1	0,08	9955,09	10/10	1,09	9976,77	10/10
PT3RD1EP2C2	0,08	1789,21	10/10	0,80	1796,42	10/10
PT3RD2EP1C1	0,28	7602,19	10/10	11,43	9565,40	10/10
PT3RD2EP1C2	0,43	1449,89	10/10	13,36	1704,02	10/10
PT3RD2EP2C1	0,11	9507,04	10/10	4,74	9584,17	10/10
PT3RD2EP2C2	0,11	1664,43	10/10	4,32	1685,02	10/10
PT3RD3EP1C1	0,23	9479,97	10/10	3,54	10686,76	10/10
PT3RD3EP1C2	0,24	1819,86	10/10	3,96	1872,32	10/10
PT3RD3EP2C1	0,20	9486,99	10/10	5,75	9992,20	10/10
PT3RD3EP2C2	0,19	1740,40	10/10	6,03	1770,43	10/10
PT3RD4EP1C1	0,32	7351,58	10/10	42,02	7346,69	10/10
PT3RD4EP1C2	0,32	1423,17	10/10	49,85	1465,10	10/10
PT3RD4EP2C1	0,15	8882,00	10/10	12,85	8882,00	10/10
PT3RD4EP2C2	0,15	1686,17	10/10	15,09	1686,17	10/10
PT3RD5EP1C1	0,19	8986,86	10/10	4,89	9670,37	10/10
PT3RD5EP1C2	0,22	1610,27	10/10	5,02	1802,29	10/10
PT3RD5EP2C1	0,12	8785,67	10/10	0,56	8779,02	10/10
PT3RD5EP2C2	0,13	1642,48	10/10	0,40	1642,08	10/10

Table 5.11: The results of the heuristic methods on the instances with size 100x8

Parameters	GRANULARITY			R/D WINDOWS		
	t[s]	Obj	Feas	t[s]	Obj	Feas
PT1RD1EP1C1	0,25	32721,57	10/10	4,70	32991,77	10/10
PT1RD1EP1C2	0,29	6213,14	10/10	3,95	6236,08	10/10
PT1RD1EP2C1	0,23	27753,53	10/10	2,72	27975,09	10/10
PT1RD1EP2C2	0,18	5070,00	10/10	2,46	5067,97	10/10
PT1RD2EP1C1	0,82	29344,92	10/10	14,58	32235,39	10/10
PT1RD2EP1C2	0,65	5530,29	10/10	8,57	5745,60	10/10
PT1RD2EP2C1	0,67	26511,50	10/10	39,36	27156,08	10/10
PT1RD2EP2C2	0,47	5004,57	10/10	-	-	9/10
PT1RD3EP1C1	0,55	30950,07	10/10	5,86	31788,18	10/10
PT1RD3EP1C2	0,50	6188,87	10/10	6,28	6218,68	10/10
PT1RD3EP2C1	0,34	26639,37	10/10	2,67	27062,24	10/10
PT1RD3EP2C2	0,30	5106,92	10/10	2,72	5091,59	10/10
PT1RD4EP1C1	0,79	23421,52	10/10	36,44	32617,82	10/10
PT1RD4EP1C2	0,84	5112,95	10/10	68,90	6175,04	10/10
PT1RD4EP2C1	0,65	25057,76	10/10	25,91	26941,22	10/10
PT1RD4EP2C2	0,72	4641,04	10/10	24,82	4881,26	10/10
PT1RD5EP1C1	0,51	28590,26	10/10	11,17	32328,81	10/10
PT1RD5EP1C2	0,46	5496,33	10/10	16,50	5971,58	10/10
PT1RD5EP2C1	0,53	25755,71	10/10	6,40	25932,56	10/10
PT1RD5EP2C2	0,53	4834,03	10/10	8,41	4841,76	10/10
PT2RD1EP1C1	0,22	41101,48	10/10	5,21	43536,36	10/10
PT2RD1EP1C2	0,26	7417,61	10/10	8,38	7716,16	10/10
PT2RD1EP2C1	0,19	32135,72	10/10	3,93	32599,53	10/10
PT2RD1EP2C2	0,13	5908,10	10/10	2,86	5925,84	10/10
PT2RD2EP1C1	0,41	34960,40	10/10	8,09	38677,49	10/10
PT2RD2EP1C2	0,66	6853,60	10/10	11,47	7280,20	10/10
PT2RD2EP2C1	0,45	31489,04	10/10	18,33	32402,72	10/10
PT2RD2EP2C2	0,41	5583,38	10/10	16,39	5600,80	10/10
PT2RD3EP1C1	0,32	38983,66	10/10	6,52	41971,34	10/10
PT2RD3EP1C2	0,43	7167,93	10/10	10,26	7414,52	10/10
PT2RD3EP2C1	0,23	30964,80	10/10	4,10	31707,31	10/10
PT2RD3EP2C2	0,22	5891,32	10/10	3,67	5909,49	10/10
PT2RD4EP1C1	0,56	28416,28	10/10	30,52	37543,05	10/10
PT2RD4EP1C2	0,61	6268,70	10/10	25,95	7083,81	10/10
PT2RD4EP2C1	0,62	29711,77	10/10	33,95	31790,29	10/10
PT2RD4EP2C2	0,44	5511,83	10/10	28,83	5700,41	10/10
PT2RD5EP1C1	0,31	33920,04	10/10	5,52	39353,14	10/10
PT2RD5EP1C2	0,34	6892,49	10/10	11,71	7692,14	10/10
PT2RD5EP2C1	0,33	30358,91	10/10	13,11	31794,77	10/10
PT2RD5EP2C2	0,29	5891,33	10/10	11,87	5998,42	10/10
PT3RD1EP1C1	0,13	16367,05	10/10	0,88	16245,14	10/10
PT3RD1EP1C2	0,11	2754,20	10/10	0,55	2710,41	10/10
PT3RD1EP2C1	0,10	13759,01	10/10	0,91	13785,77	10/10
PT3RD1EP2C2	0,12	2481,36	10/10	1,44	2494,61	10/10
PT3RD2EP1C1	0,30	10917,75	10/10	19,49	12573,87	10/10
PT3RD2EP1C2	0,33	2050,37	10/10	29,40	2236,16	10/10
PT3RD2EP2C1	0,16	12351,06	10/10	8,79	12649,22	10/10
PT3RD2EP2C2	0,16	2160,21	10/10	8,59	2205,39	10/10
PT3RD3EP1C1	0,27	13287,04	10/10	16,68	13853,73	10/10
PT3RD3EP1C2	0,28	2486,95	10/10	8,57	2412,14	10/10
PT3RD3EP2C1	0,28	12729,42	10/10	12,09	13015,84	10/10
PT3RD3EP2C2	0,28	2307,95	10/10	9,44	2352,37	10/10
PT3RD4EP1C1	0,48	10111,22	10/10	49,14	12114,86	10/10
PT3RD4EP1C2	0,47	1827,32	10/10	63,43	2127,50	10/10
PT3RD4EP2C1	0,22	12003,79	10/10	20,77	12403,60	10/10
PT3RD4EP2C2	0,22	2216,99	10/10	22,50	2301,85	10/10
PT3RD5EP1C1	0,28	11645,60	10/10	7,03	13605,98	10/10
PT3RD5EP1C2	0,20	2210,45	10/10	7,83	2517,46	10/10
PT3RD5EP2C1	0,26	12114,76	10/10	3,28	12155,02	10/10
PT3RD5EP2C2	0,25	2220,53	10/10	3,18	2222,44	10/10

Table 5.12: The results of the heuristic methods on the instances with size 135x10

Parameters	GRANULARITY			R/D WINDOWS		
	t[s]	Obj	Feas	t[s]	Obj	Feas
PT1RD1EP1C1	0,41	42641,26	10/10	6,80	42845,12632	10/10
PT1RD1EP1C2	0,42	8073,90	10/10	6,67	8047,92655	10/10
PT1RD1EP2C1	0,26	34451,69	10/10	3,82	34666,9055	10/10
PT1RD1EP2C2	0,21	6397,75	10/10	3,32	6384,589183	10/10
PT1RD2EP1C1	0,81	36797,64	10/10	13,30	40106,5309	10/10
PT1RD2EP1C2	0,88	7199,95	10/10	14,36	7264,6053	10/10
PT1RD2EP2C1	0,83	32616,72	10/10	38,21	33351,86083	10/10
PT1RD2EP2C2	0,64	6011,67	10/10	33,50	6006,176633	10/10
PT1RD3EP1C1	0,62	41100,16	10/10	8,09	43008,16303	10/10
PT1RD3EP1C2	0,54	7901,72	10/10	10,55	7794,630233	10/10
PT1RD3EP2C1	0,42	33633,49	10/10	3,30	34166,55352	10/10
PT1RD3EP2C2	0,42	6363,97	10/10	3,55	6337,576583	10/10
PT1RD4EP1C1	0,96	32083,85	10/10	60,00	41334,16368	10/10
PT1RD4EP1C2	1,06	6643,14	10/10	60,00	7943,924267	10/10
PT1RD4EP2C1	1,32	31590,63	10/10	37,54	33932,08388	10/10
PT1RD4EP2C2	1,27	5736,88	10/10	32,37	6023,420267	10/10
PT1RD5EP1C1	0,56	37408,32	10/10	8,89	43371,63648	10/10
PT1RD5EP1C2	0,65	6944,17	10/10	15,08	7500,4807	10/10
PT1RD5EP2C1	1,03	32892,02	10/10	12,44	33834,20015	10/10
PT1RD5EP2C2	0,63	6112,44	10/10	10,98	6091,88075	10/10
PT2RD1EP1C1	0,26	48814,70	10/10	11,59	53039,38248	10/10
PT2RD1EP1C2	0,35	9861,55	10/10	16,27	9898,15	10/10
PT2RD1EP2C1	0,21	41208,58	10/10	7,75	41796,46152	10/10
PT2RD1EP2C2	0,21	7539,47	10/10	5,64	7582,452717	10/10
PT2RD2EP1C1	0,53	46482,52	10/10	11,98	49983,9904	10/10
PT2RD2EP1C2	0,57	8799,79	10/10	10,30	9029,48645	10/10
PT2RD2EP2C1	0,60	40182,06	10/10	33,77	41115,32268	10/10
PT2RD2EP2C2	0,52	7558,62	10/10	33,07	7580,3828	10/10
PT2RD3EP1C1	0,44	47216,03	10/10	9,07	50872,30357	10/10
PT2RD3EP1C2	0,46	9155,60	10/10	11,29	9402,7514	10/10
PT2RD3EP2C1	0,30	41261,55	10/10	5,32	41980,75987	10/10
PT2RD3EP2C2	0,32	7549,15	10/10	5,97	7582,740083	10/10
PT2RD4EP1C1	0,71	39823,85	10/10	37,36	50512,6505	10/10
PT2RD4EP1C2	0,77	8041,83	10/10	32,12	8875,9215	10/10
PT2RD4EP2C1	0,86	38710,66	10/10	31,61	41597,001	10/10
PT2RD4EP2C2	0,59	7060,56	10/10	30,35	7252,772	10/10
PT2RD5EP1C1	0,50	44702,69	10/10	12,64	53911,2258	10/10
PT2RD5EP1C2	0,52	8774,05	10/10	19,81	9756,694767	10/10
PT2RD5EP2C1	0,44	38640,79	10/10	22,56	40499,49473	10/10
PT2RD5EP2C2	0,37	7263,38	10/10	19,98	7445,727	10/10
PT3RD1EP1C1	0,19	19688,49	10/10	1,73	19416,93813	10/10
PT3RD1EP1C2	0,19	3568,16	10/10	1,07	3506,03385	10/10
PT3RD1EP2C1	0,14	17111,02	10/10	2,00	17219,30007	10/10
PT3RD1EP2C2	0,13	3132,04	10/10	2,50	3162,7077	10/10
PT3RD2EP1C1	0,36	13619,22	10/10	44,65	15590,48157	10/10
PT3RD2EP1C2	0,59	2536,12	10/10	-	-	9/10
PT3RD2EP2C1	0,28	15063,06	10/10	12,94	15098,6644	10/10
PT3RD2EP2C2	0,25	2800,97	10/10	12,09	2809,30265	10/10
PT3RD3EP1C1	0,39	15709,44	10/10	9,65	16754,04597	10/10
PT3RD3EP1C2	0,35	3218,21	10/10	8,57	3197,070817	10/10
PT3RD3EP2C1	0,33	16652,45	10/10	15,66	17022,66287	10/10
PT3RD3EP2C2	0,33	2980,71	10/10	15,36	2974,408567	10/10
PT3RD4EP1C1	0,60	12904,05	10/10	60,00	15902,73298	10/10
PT3RD4EP1C2	0,63	2412,50	10/10	60,00	2938,321067	10/10
PT3RD4EP2C1	0,27	15743,77	10/10	13,43	15981,96275	10/10
PT3RD4EP2C2	0,27	2818,62	10/10	15,68	2887,423783	10/10
PT3RD5EP1C1	0,44	14980,40	10/10	13,61	16999,32567	10/10
PT3RD5EP1C2	0,26	2959,92	10/10	15,33	3087,774317	10/10
PT3RD5EP2C1	0,33	15750,41	10/10	7,24	15953,96627	10/10
PT3RD5EP2C2	0,31	2879,81	10/10	7,85	2969,237183	10/10

Table 5.13: The results of the heuristic methods on the instances with size 170x12

Parameters	GRANULARITY			R/D WINDOWS		
	t[s]	Obj	Feas	t[s]	Obj	Feas
PT1RD1EP1C1	0,29	48943,10	10/10	6,33	49623,81	10/10
PT1RD1EP1C2	0,29	9310,25	10/10	7,53	9218,10	10/10
PT1RD1EP2C1	0,25	40770,22	10/10	5,95	41032,97	10/10
PT1RD1EP2C2	0,23	7465,44	10/10	4,77	7445,99	10/10
PT1RD2EP1C1	0,75	41758,91	10/10	20,80	45134,56	10/10
PT1RD2EP1C2	0,97	8411,30	10/10	14,43	8461,04	10/10
PT1RD2EP2C1	1,08	38914,68	10/10	-	-	7/10
PT1RD2EP2C2	0,69	7384,45	10/10	60,00	7328,45	10/10
PT1RD3EP1C1	0,69	46655,92	10/10	10,53	48515,33	10/10
PT1RD3EP1C2	0,74	8818,38	10/10	8,80	8782,26	10/10
PT1RD3EP2C1	0,44	38919,04	10/10	4,53	39682,40	10/10
PT1RD3EP2C2	0,42	7608,74	10/10	4,82	7578,23	10/10
PT1RD4EP1C1	1,48	35550,03	10/10	27,11	49587,18	10/10
PT1RD4EP1C2	1,30	7569,85	10/10	51,34	9256,46	10/10
PT1RD4EP2C1	1,47	37919,27	10/10	32,87	41644,63	10/10
PT1RD4EP2C2	1,67	6900,09	10/10	22,69	7206,42	10/10
PT1RD5EP1C1	0,69	40714,55	10/10	8,01	47079,44	10/10
PT1RD5EP1C2	0,78	8085,86	10/10	12,09	8799,11	10/10
PT1RD5EP2C1	0,79	38473,43	10/10	13,76	38960,69	10/10
PT1RD5EP2C2	0,63	7087,68	10/10	16,72	7053,40	10/10
PT2RD1EP1C1	0,29	58132,35	10/10	8,46	61241,01	10/10
PT2RD1EP1C2	0,34	11321,26	10/10	11,69	11598,93	10/10
PT2RD1EP2C1	0,21	48356,71	10/10	4,86	48889,15	10/10
PT2RD1EP2C2	0,21	8899,84	10/10	6,00	8938,21	10/10
PT2RD2EP1C1	0,57	52406,30	10/10	14,71	57347,18	10/10
PT2RD2EP1C2	0,65	9797,59	10/10	13,04	10119,96	10/10
PT2RD2EP2C1	0,67	44412,85	10/10	25,65	45884,22	10/10
PT2RD2EP2C2	0,60	8436,33	10/10	26,52	8442,52	10/10
PT2RD3EP1C1	0,51	54685,43	10/10	13,40	59019,76	10/10
PT2RD3EP1C2	0,50	10723,32	10/10	11,02	10894,67	10/10
PT2RD3EP2C1	0,34	46349,19	10/10	5,87	47296,25	10/10
PT2RD3EP2C2	0,28	8513,65	10/10	6,26	8506,25	10/10
PT2RD4EP1C1	0,91	41849,49	10/10	44,34	56410,84	10/10
PT2RD4EP1C2	0,81	9585,06	10/10	46,87	10760,87	10/10
PT2RD4EP2C1	0,80	44406,95	10/10	42,22	48335,41	10/10
PT2RD4EP2C2	0,59	8469,35	10/10	40,81	8810,14	10/10
PT2RD5EP1C1	0,51	47350,32	10/10	16,22	59352,40	10/10
PT2RD5EP1C2	0,64	9878,30	10/10	13,68	11279,06	10/10
PT2RD5EP2C1	0,53	46201,07	10/10	28,70	48985,99	10/10
PT2RD5EP2C2	0,56	8526,39	10/10	27,04	8754,95	10/10
PT3RD1EP1C1	0,17	22559,94	10/10	1,12	22210,40	10/10
PT3RD1EP1C2	0,18	4193,67	10/10	1,24	4130,21	10/10
PT3RD1EP2C1	0,12	19846,71	10/10	2,18	19939,91	10/10
PT3RD1EP2C2	0,12	3626,19	10/10	1,77	3663,73	10/10
PT3RD2EP1C1	0,49	15490,83	10/10	59,66	16385,02	10/10
PT3RD2EP1C2	0,38	2948,84	10/10	-	-	3/10
PT3RD2EP2C1	0,23	18428,10	10/10	12,85	18428,10	10/10
PT3RD2EP2C2	0,20	3282,71	10/10	16,77	3282,71	10/10
PT3RD3EP1C1	0,37	19184,49	10/10	40,97	19675,56	10/10
PT3RD3EP1C2	0,38	3830,28	10/10	16,00	3738,00	10/10
PT3RD3EP2C1	0,38	19139,98	10/10	24,17	19502,85	10/10
PT3RD3EP2C2	0,38	3404,68	10/10	23,90	3377,67	10/10
PT3RD4EP1C1	0,66	15192,45	10/10	38,09	18271,78	10/10
PT3RD4EP1C2	0,68	2805,60	10/10	40,68	3031,95	10/10
PT3RD4EP2C1	0,32	18086,75	10/10	7,93	18101,83	10/10
PT3RD4EP2C2	0,32	3394,31	10/10	6,53	3395,89	10/10
PT3RD5EP1C1	0,34	17727,07	10/10	38,69	19227,11	10/10
PT3RD5EP1C2	0,33	3312,20	10/10	19,55	3467,79	10/10
PT3RD5EP2C1	0,35	18697,08	10/10	9,82	18615,90	10/10
PT3RD5EP2C2	0,34	3341,89	10/10	9,60	3328,11	10/10

Table 5.14: The results of the heuristic methods on the instances with size 200x15

Instance size	GRANULARITY			R/D WINDOWS		
	t[s]	Obj	Feas	t[s]	Obj	Feas
100x8	0,27	10636,77	60/60	9,28	11369,32	60/60
135x10	0,38	14718,16	60/60	14,36	15778,64	59/60
170x12	0,50	18902,41	60/60	18,14	20223,74	59/60
200x15	0,54	21417,12	60/60	19,17	23017,81	58/60

Table 5.15: The aggregated results of the heuristic methods on large instances

5.3 Usage in practice

The methods designed in this work can be used for scheduling the industry processes in energy demanding production stage. For example, it can be used in a production of toughened glass, where the jobs are heating of glass panels in furnaces. The material, which comes from previous stages (e.g. some store) can represent the jobs scheduled in IPMSMEC problem. To integrate scheduling of the energy-demanding stage to other stages release times and deadlines can be used. The release times are defined by the times when the jobs are available after finishing the processing of them in a previous stage. The deadlines can be defined by the times when the material represented by the job processed in current stage is needed to be sent to next stage. In a case of the melting, the processing times and energy consumptions of jobs can be affected and defined by the amount and chemical properties of the processed material.

5.3.1 Plant Simulation

The Plant Simulation [21] tool can be used for modelling and simulation of our melting production stage. The tool allows users to create digital models of production systems so that the characteristics of the system can be explored and the performance can be optimized before it is actually constructed. It provides analysis tools such as bottleneck analysis, statistics and charts so that the user can evaluate different manufacturing scenarios.

For our purpose, we need to provide some input data to the tool. It can be achieved by SimTalk programming language described in [2] together with all the objects provided by Plant Simulation and with usage examples. The 3D model of small production stage is provided in Figure 5.1

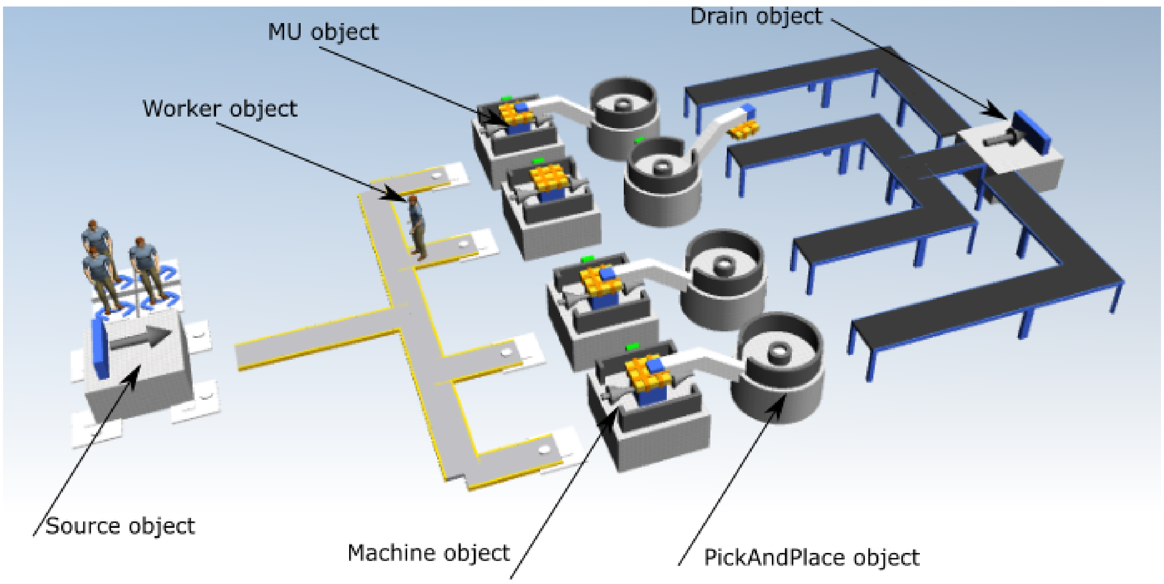


Figure 5.1: Model of production stage

The batch of material for melting (job) is represented by the MU object and provided by the Source object in times according to DeliveryTable object, which specifies MU's attributes (i.e., start times, energy consumptions, processing times, allocated machines, etc.). The attributes of MU objects are based on the schedule provided by our exact methods solving the IPMSMEC problem so that the jobs are processed according to this schedule. The MU is transported by the Worker object to the appropriate Machine object, which is allocated for it. The Machine object starts to process the MU according to the start time and consumes electricity according to the predefined consumption of the MU. After the time period represented by the processing time, the processed MU object is picked by the PickAndPlace object, which sends it to the Line object that transports it to the Drain object. At that moment, the job represented by the MU object is ready to be processed by next production stage of the factory.

After the modelling, we can mediate the parameters of the jobs and the output values of the method solving the IPMSMEC problem and start the simulation. The instance simulated by our model consists of 80 jobs and 4 machines. The parameters of the jobs and the output values provided by solving the instance using the Two-Phase method are provided in the Table 5.16.

The Plant Simulation provides a tool for electricity consumption analysis. After the simulation, the chart that presents an amount of consumed electricity in dependence on time is generated. The chart together with the prices of electricity used in the solved instance is presented in Figure 5.2. It shows that Two-Phase method schedules the jobs so that the electricity is not consumed in the periods with the highest electricity price.

Job[j]	Input parameters				Output parameters	
	p_j	r_j	d_j	c_j	s_j	m
1	38	1	1440	4	1196	1
2	61	1	1440	7	1317	4
3	103	1	1440	2	720	3
4	2	1	1440	6	1320	2
5	20	1	1440	4	1000	4
6	49	1	1440	8	1386	2
7	85	1	1440	5	335	2
8	1	1	1440	7	1319	2
9	12	1	1440	3	1002	2
10	50	1	1440	3	1140	1
11	111	1	1440	7	1	1
12	16	1	1440	10	1424	4
13	89	1	1440	10	131	4
14	15	1	1440	7	300	1
15	7	1	1440	3	993	4
16	97	1	1440	5	323	3
17	82	1	1440	4	793	1
18	90	1	1440	2	901	4
19	34	1	1440	7	315	1
20	63	1	1440	1	659	1
21	6	1	1440	4	1014	2
22	7	1	1440	2	1191	4
23	68	1	1440	2	720	4
24	55	1	1440	1	420	1
25	46	1	1440	5	1260	3
26	66	1	1440	4	1194	3
27	66	1	1440	2	1140	2
28	53	1	1440	10	1387	3
29	37	1	1440	8	263	1
30	26	1	1440	3	755	2
31	3	1	1440	4	1206	2
32	110	1	1440	5	1209	2
33	24	1	1440	7	311	2
34	113	1	1440	3	788	4
35	109	1	1440	6	1	2
36	95	1	1440	1	660	2
37	32	1	1440	8	110	2
38	14	1	1440	2	1140	3
39	42	1	1440	8	230	2
40	105	1	1440	2	897	2
41	68	1	1440	9	180	3
42	103	1	1440	5	1214	4
43	75	1	1440	8	248	3
44	16	1	1440	4	1198	4
45	6	1	1440	3	1190	1
46	55	1	1440	6	1331	2
47	63	1	1440	5	357	4
48	2	1	1440	4	991	4
49	3	1	1440	10	142	2
50	71	1	1440	5	349	1
51	116	1	1440	4	781	2
52	55	1	1440	8	65	3
53	5	1	1440	10	1435	2
54	55	1	1440	9	1383	1
55	59	1	1440	3	961	1
56	32	1	1440	3	823	3
57	25	1	1440	8	106	4
58	81	1	1440	6	1306	3
59	60	1	1440	10	120	3
60	71	1	1440	2	722	1
61	116	1	1440	8	220	4
62	47	1	1440	8	216	1
63	86	1	1440	2	875	1
64	39	1	1440	8	272	2
65	40	1	1440	1	1154	3
66	56	1	1440	4	964	3
67	104	1	1440	9	112	1
68	2	1	1440	10	1438	1
69	64	1	1440	7	1	3
70	14	1	1440	2	1177	4
71	1	1	1440	2	1176	4
72	21	1	1440	7	336	4
73	40	1	1440	5	1234	1
74	36	1	1440	3	1140	4
75	105	1	1440	7	1	4
76	85	1	1440	9	145	2
77	46	1	1440	8	1378	4
78	9	1	1440	6	1322	2
79	109	1	1440	2	855	3
80	109	1	1440	6	1274	1

Table 5.16: Input parameters for Plant Simulation model

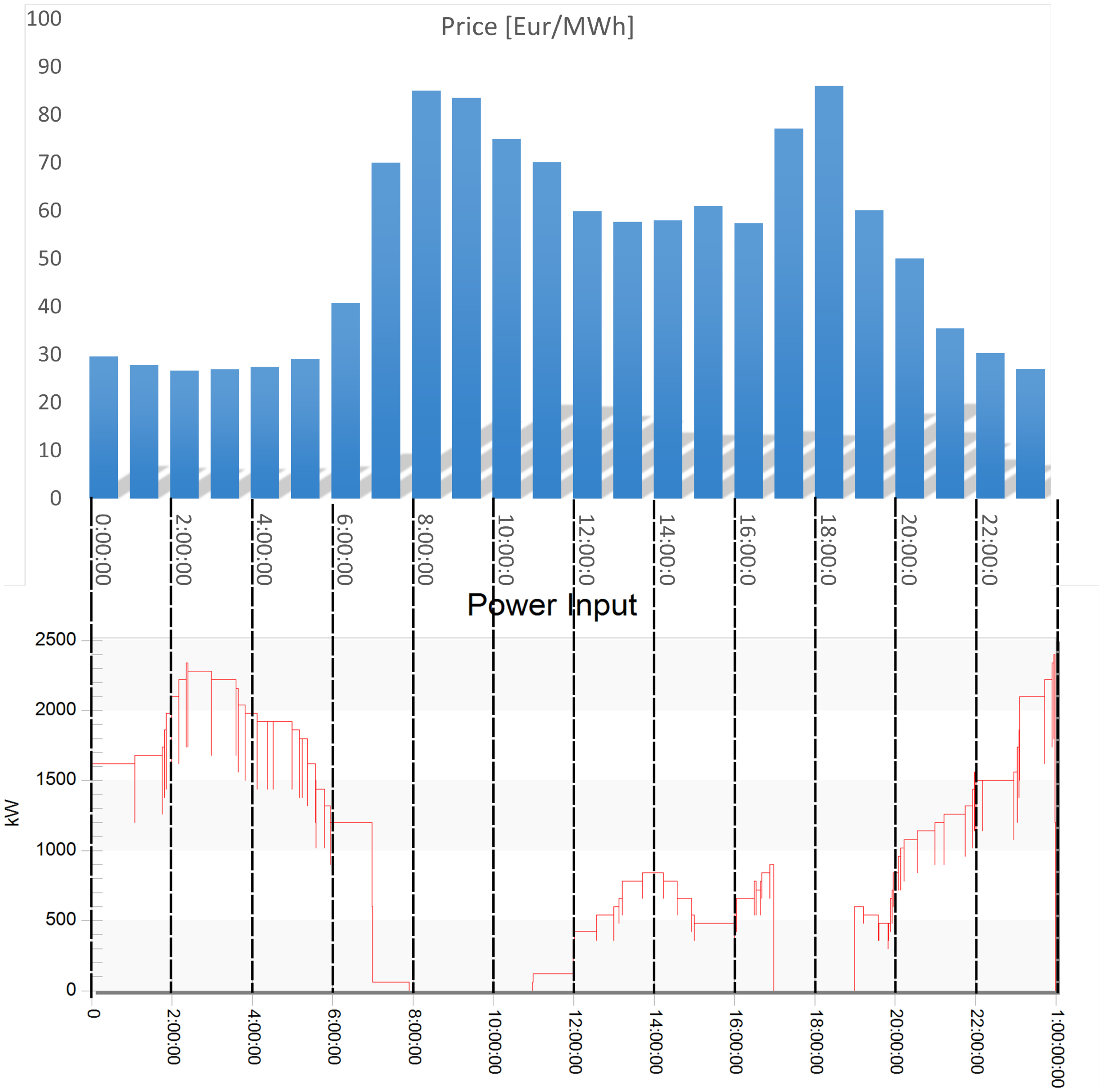


Figure 5.2: Electricity consumption by the simulation of problem instance

Chapter 6

Conclusion

This master thesis deals with the NP-hard problem of **Identical Parallel Machine Scheduling Minimizing Electricity Costs** (**IPMSMEC**). Firstly, the motivation for solving IPMSMEC problem including the situation on Czech electricity market is presented. After that, the relevant literature is reviewed, and the IPMSMEC problem addressed in this work is stated together with its complexity. It is followed by designing of four exact methods of which two are the standard methods inspired by the literature and modified to fit our problem. The remaining two methods are fully our contribution. The exact methods are followed by the designing of two heuristics used for solving large instances of our problem that can not be solved by the exact methods in an acceptable time. It is followed by results of the experiments performed on all the methods. The method for generating a lot of test instances for IPMSMEC problem is also provided in the thesis. It is also presented that the methods for solving the IPMSMEC problem can be used for scheduling the processes of specific stages of many production enterprises. The standardized tool was used for modelling and simulation of the exemplary production stage with a usage of a schedule provided by the exact methods as an input. It proved that the methods could minimize the costs of the electricity consumption by scheduling the energy-demanding jobs into the time periods, in which the price of electricity is lower than in the others.

The experiments showed that it is very hard to beat the state-of-the-art MILP solvers by the complex algorithms, which are trying to intelligently restrict or search through the solution space (i.e. Logic Based Benders Decomposition method). Instead of that, it is important to try to exploit some key properties of addressed problem. The described approach was crucial for the Two-phase method, which is based on the decomposition of standard time-indexed formulation into two phases. As illustrated by the results of the experiments, the Two-Phase method proved to be the most effective of all the exact methods presented in this work, and it is our main contribution. The effectiveness of the Two-Phase method is also documented by the stunning results of the heuristic method which restricts the space of possible start times to be a multiple of 15 minutes and then uses Two-Phase method for finding the optimal start times in this reduced solution space.

The experiments were performed using the instances with sizes based on the number of jobs (set J) and machines (set M) for scheduling. The largest instances on which the exact methods were tested had size $|J| = 60, |M| = 12$ and the most efficient Two-phase method needed $6,12s$ in average for finding the optimal solution. The previously mentioned heuristic method, which restricts the space of possible start times to be a multiple of 15 minutes needed $0,54s$ in average for finding the optimal solution for instances with size $|J| = 200, |M| = 15$.

Future work involves the improvement of the second heuristic method proposed in this work (Changing release time/deadline time-windows), which tries to intelligently assign the jobs to the time periods and change the release times and deadlines of the jobs based on the assignment. The results of the experiments proved that the heuristic was not able to solve few of the largest test instances in the time limit and in a comparison with the second heuristic, it proved to be much less effective.

Bibliography

[1] S Ashok. Peak-load management in steel plants. *Applied energy*, 83(5):413–424, 2006.

[2] Steffen Bangsow. *Manufacturing simulation with plant simulation and simtalk: usage and programming with examples and solutions*. Springer Science & Business Media, 2010.

[3] Philippe Baptiste, Jacques Carlier, and Antoine Jouglet. A branch-and-bound procedure to minimize total tardiness on one machine with arbitrary release dates. *European Journal of Operational Research*, 158(3):595 – 608, 2004.

[4] Peter Brucker. *Scheduling algorithms*. Springer-Verlag Berlin Heidelberg, 5th ed edition, 2007.

[5] Pedro M Castro, Iiro Harjunkoski, and Ignacio E Grossmann. New continuous-time scheduling formulation for continuous plants under variable electricity cost. *Industrial & engineering chemistry research*, 48(14):6701–6714, 2009.

[6] Pedro M Castro, Iiro Harjunkoski, and Ignacio E Grossmann. Optimal scheduling of continuous plants with energy constraints. *Computers & Chemical Engineering*, 35(2):372–387, 2011.

[7] J. Y. Ding, S. Song, R. Zhang, R. Chiong, and C. Wu. Parallel machine scheduling under time-of-use electricity prices: New models and optimization approaches. *IEEE Transactions on Automation Science and Engineering*, 13(2):1138–1154, April 2016.

[8] Paul E. Dunne. An annotated list of selected np-complete problems. http://cgi.csc.liv.ac.uk/~ped/teachadmin/COMP202/annotated_np.html. Accessed: 2017-16-5.

[9] Asociace energetických manažerů. *Trh s elektřinou: úvod do liberalizované energetiky*. Asociace energetických manažerů, vydání druhé, aktualizované edition, 2016.

[10] Dragoljub Gajic, Hubert Hadera, Luca Onofri, Iiro Harjunkoski, and Stefano Di Gennaro. Implementation of an integrated production and electricity optimization system in melt shop. *Journal of Cleaner Production*, 2016.

[11] Hubert Hadera, Iiro Harjunkoski, Guido Sand, Ignacio E Grossmann, and Sebastian Engell. Optimization of steel production scheduling with complex time-sensitive electricity cost. *Computers & Chemical Engineering*, 76:117–136, 2015.

[12] Alain Hait and Christian Artigues. A hybrid cp/milp method for scheduling with energy costs. *European Journal of Industrial Engineering*, 5(4):471–489, 2011.

[13] Alain Haït and Christian Artigues. On electrical load tracking scheduling for a steel plant. *Computers & Chemical Engineering*, 35(12):3044 – 3047, 2011.

[14] John N Hooker. Planning and scheduling by logic-based benders decomposition. *Operations Research*, 55(3):588–602, 2007.

[15] Sumit Mitra, Ignacio E Grossmann, Jose M Pinto, and Nikhil Arora. Optimal production planning under time-sensitive electricity prices for continuous power-intensive processes. *Computers & Chemical Engineering*, 38:171–184, 2012.

[16] OTE. Denní trh. http://www.ote-cr.cz/kratkodobe-trhy/elektrina/denni-trh, 2010. Accessed: 2017-16-3.

[17] OTE. Základní údaje. http://www.ote-cr.cz/o-spolecnosti/zakladni-udaje, 2010. Accessed: 2016-10-9.

[18] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[19] Maximilian Selmair, Thorsten Claus, Marco Trost, Andreas Bley, and Frank Herrmann. Job shop scheduling with flexible energy prices. In *European Council for Modeling and Simulation*, 2016.

[20] Fadi Shrouf, Joaquin Ordieres-Meré, Alvaro García-Sánchez, and Miguel Ortega-Mier. Optimizing the production scheduling of a single machine to minimize total energy consumption costs. *Journal of Cleaner Production*, 67:197–207, 2014.

[21] Siemens Industry Software. Plant simulation. http://www.plm.automation.siemens.com/en_gb/products/tecnomatix/manufacturing-simulation/material-flow/plant-simulation.shtml, 2017. Accessed: 2017-21-4.

[22] Yuan-yuan TAN, Ying-lei HUANG, and Shi-xin LIU. Two-stage mathematical programming approach for steelmaking process scheduling under variable electricity price. *Journal of Iron and Steel Research, International*, 20(7):1–8, 2013.

[23] David Vobořil. Trh s elektřinou. http://oenergetice.cz/elektrina/trh-s-elektrinou/trh-s-elektrinou, 2015. Accessed: 2016-10-9.

[24] Qi Zhang and Ignacio E Grossmann. Planning and scheduling for industrial demand side management: advances and challenges. In *Alternative Energy Sources and Technologies*, pages 383–414. Springer, 2016.