A Hybrid Solver for Maximizing the Profit of an Energy Company

Łukasz Domagała, Tomasz Wojdyła, Wojciech Legierski and Michał Swiderski
Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, 16 Akademicka Str., 44-100 Gliwice, Poland

Keywords: Optimization, Profit, Energy, Industry.

Abstract: An energy company manages power stations, handles sales and purchases of electrical energy, CO2 emission permits and other goods. The goal of such a company is to ensure energy safety of its clients and maximize the profit. The problem is complex because of its structure and size therefore efficient automated approaches for solving it are in demand. We have generalized the problem definition to account for any structure of the power stations, market data and time scope. The definition describes a non-linear combinatorial optimization problem. We have tested a number of approaches including: constraint/ logic/ dynamic/ integer/ linear programming, local search and their hybrids using prototypes with input data from a real life process. We present a hybrid solver to produce an acceptable, near optimal solution which satisfies the requirements of an industrial application. Our research is a road-sign for development of similar software for the energy industry.

1 INTRODUCTION

The aim of the solver is to return a schedule of production, sales and purchases of all the goods for a given *time horizon* that satisfies all the *hard constraints* and optimizes the *objective function*. Hard constraints are these that cannot be violated in the solution. Satisfying them guarantees that technological and marketing requirements are met and ensures the energy safety of the company's clients. The objective function is the company's profit gained for a given time horizon. Maximizing this profit is the main goal. The optimization is performed for problem instances which consist of: technological capabilities and parameters of the power stations, market plans and estimates supplied by marketing and financial experts, trade contracts, initial states for those goods that can be accumulated.

2 PROBLEM DEFINITION

We have obtained instances of the profit maximization problem during work on a commercial project. Basing on the problem instances, we have built a generalized problem definition that accounts for any structure of the power stations, market data and time scope. To the best knowledge of the authors a definition of such a problem has never been published before.

2.1 Timing and Notation

The production, trade and constraint setup is performed for discrete time *periods*. $h \in [1,H]$ is the shortest period called, for convenience, an *hour*. Each value of h is categorized as *peak* or *off-peak*. Furthermore, consecutive values of h are grouped into periods $(H_{m-1},H_m] = M_m$ indexed by $m \in [1,M]$ where $H_0 = 0, H_{m-1} < H_m, H_M = H$, which are, for convenience, called *months*. The period of $[1,H] = y$ is, for convenience, called a *year*. An energy company handles the following *goods*: energy $\{en\}$, CO2 emission permits $\{ep_p, p \in [1,P]\}$, financial benefits $\{be_b, b \in [1,B]\}$. An energy company handles the following *objects*: power stations, production units, sales, purchases. Some object-period pairs have corresponding control variables $v(object, period)$. Numerical (unless otherwise stated) attributes $\alpha(type, object/good, period)$ are attached to objects/goods, where *period* denotes the period to which it applies. The term "volume" is used to describe the quantity of some good.

2.2 Energy Production and Trade

An *energy company* (*ec*) is divided into *power stations* $\{ps_s : s \in [1,S]\}$. Each power station ps_s is divided into *production units* $\{pu_i^s : i \in [1,U^s]\}$ which produce (electrical) *energy*. *ec* manages the volumes

Domagała Ł., Wojdyła T., Legierski W. and Świderski M..
A Hybrid Solver for Maximizing the Profit of an Energy Company.
DOI: 10.5220/0004000602780283
In *Proceedings of the 14th International Conference on Enterprise Information Systems* (ICEIS-2012), pages 278-283
ISBN: 978-989-8565-10-5

$vps_{s,h}$ of energy supplied by ps_s in period h. $vps_{s,h} = fpv_{s,h}(\sum_{i=1}^{U^s} v(pu_i^s,h))$ where $fpv_{s,h}$ is a *piecewise linear function*. $fpv_{s,h}$ includes a number of components:

- The actions of a regulatory body, which may intervene with the production plans and are meant to regulate the energy market. These actions are predicted by experts as a piecewise linear function $frp_{s,h}(\sum_{i=1}^{U^s} v(pu_i^s,h))$.
- The error factor $\alpha(zo,ps_s,h)$ associated with the imperfections of the energy distribution network
- The sales of energy $\{v(se_i^s,h) : i \in [1,Se^s]\}$ managed privately by the ps_s

The energy trade consists of purchases $\{ze_i : i \in [1,Ze]\}$ and sales $\{se_i : i \in [1,Se]\}$ managed by the *ec*. The trade is further divided into *contracts* and *plans*. The contract is a signed trade agreement, whereas the trade plan is based on the expert predictions. This distinction, however, is reflected in the *variable domains* and is transparent for the solver.

2.3 CO2 Emission Permits

The CO2 emissions of the ps_s have to be covered by *permits* of $\{ep_p, p \in [1,P]\}$ types. Permits may be traded, may be granted by the government, may be *consumed*, are limited by constraints depending on the ep_p. The permit trade is managed for each ps_s separately. $\forall p \in [1,P], s \in [1,S]$ the defined sales and purchases are respectively $\{sp_i^{p,s} : i \in [1,Sp^{p,s}]\}$, $\{zp_i^{p,s} : i \in [1,Zp^{p,s}]\}$. The permits are consumed to cover emissions which are relative to energy production. Consumption volume is $v(pu_i^s,h) \cdot \alpha(co2,pu_i^s,h)$, where $\alpha(co2,pu_i^s,h)$, is the emission ratio.

2.4 Financial Benefits

Financial benefits of type $be_b, b \in [1,B]$ can be produced, purchased, sold or consumed. They are produced relatively to energy production, be_b production volume is $v(pu_i^s,h) \cdot \alpha(be_b,pu_i^s,h)$. The ratios of production are dependent on the efficiency of, and resources used by pu_i^s. Examples of financial benefit types are type for energy produced from renewable resources, type for high efficiency coal powered production, type for natural gas powered production, etc. Sales and purchases respectively, are denoted by $\{sb_i^b : i \in [1,Sb^b], b \in [1,B]\}$, $\{zb_i^b : i \in [1,Zb^b], b \in [1,B]\}$. The be_b is consumed, relatively to the volume of sold energy, to gain access to certain energy markets, be_b consumption volume is $v(se_i,h) \cdot \alpha(be_b,se_i,h)$.

2.5 Control Variables

The solution to the optimization problem is defined by values assigned to the control variables. The complete set CV of control variables (see **APPENDIX A**) is (A.1)-(A.7) where (A.1) are energy production variables, (A.2)-(A.3) are energy trade variables, (A.4)-(A.5) are financial benefits trade variables, (A.6)-(A.7) are CO2 emission permit trade variables.

2.6 Constraints

The formulas representing linear constraints are: variable unary (variable domains) (A.8), production gradient (A.9), a technological constraint of each pu_i^s, energy balance (A.10), financial benefit monthly balance (A.12), financial benefits yearly balance (A.11), CO2 permits nonnegativity (A.13), CO2 permits yearly balance (A.14). (A.12)-(A.14) are called *long period constraints*. The attribute name *ist* explicitly denotes the initial state, whenever indexation refers to element 0 e.g. $v(pu_i^s,0)$ it signifies an implicit initial state.

The formulas representing nonlinear constraints are: minimal duration for which a pu_i^s has to work after *startup* (A.15), technological constraint for the level of production (A.16), minimal number of pu_i^s turned on in ps_s (A.17), startup schedule of a pu_i^s (A.18), relation between the production levels of pu_i^s and energy provided to the *ec* by ps_s (A.19). Attribute $\alpha(startup,pu_i^s,y)$ is an ordered set of values, # is a set cardinal number.

2.7 Elements of the Objective Function

The objective function $\omega(CV)$ represents the total *ec* profit. Each control variable has a corresponding profit ratio represented by the $profit$ attribute. For production and purchases the profit ratio is negative and for sales the profit ratio is positive. The profit of control variables is linear and represented by (A.20). (A.21) and (A.22) are nonlinear elements of the cost function. The first represents the startup cost of pu_i^s i.e. the cost of turning on a disabled production unit. The latter corresponds to costs related to components of the $fpv_{s,h}$ (Section 2.2).

3 TESTED APPROACHES

The approaches have been tested on problem *instances* denoted by $inst(ec,H)$ where *ec* is the definition of objects and types, H is the time horizon. In particular $inst(ec^r,H^r)$ denotes the industrial

real life problem instance. $inst(ec^r, H^r)$ consists of 289'200 control variables, 411'838 linear non-unary constraints and 490'560 nonlinear constraints for the $H^r = 8'760$. Under the confidentiality agreement we are not allowed to disclose the structure of ec^r. For the time $cr(st)$ used to perform the optimization, the condition $cr(st) > 10min$ is called the $timeout$. $\neg timeout$ is a requirement for the solver.

We have used the following criteria to compare models: $cr(nlin)$ are nonlinearities included in the model, $cr(gopt)$ is guarantee of optimality provided, $cr(long)$ are long period constraints included in the model, $cr(teff) = cr(st)/H$ time efficiency. The approaches have been tested on personal computers with 2 x 2.2Ghz processors, 3GB of RAM and address space.

3.1 Constraint (Logic) Programming

Constraint programming (CP) (Apt, 2009; Marriott and Stuckey, 1998) is a programming paradigm with the central notion of a constraint. A constraint states relations between variable domains (allowed combinations of domain values). CP is a form of declarative programming where the program in the form of *constraint statements* is a description of the problem, rather than a path to the solution (unlike in the case of procedural programming). CP makes a distinction between 3 components required to obtain a solution: the declarative constraint statement, *constraint propagation* and *search*. Historically CP has grown out of, and has been embedded in *logic programming* and often uses the LP based backtracking search, it is however possible to embed constraint programming in a procedural language.

We have tested models written under two CP systems (Schulte, 2010; Szymanek and Kuchciński, 2010) and a CLP system (Cisco Systems, 2010) with the result of obtaining optimal solutions for problem instances with $H \in [1, \lfloor H^r/2000 \rfloor]$. The $cr(teff) \in [5s, 25s]$ depending on the CP system, constraint propagation methods and search methods. However for problem instances with $H \geq \lfloor H^r/100 \rfloor$ we have been unable to produce a solution without a timeout.

The advantages of the CP approach are that the complete problem model can be taken into account $cr(long) = cr(nlin) = true$, solution with the guarantee of optimality can be obtained $cr(gopt) = true$. The major disadvantage is that the models are impractical for problem instances near the real life problem size

3.2 Dynamic Programming

Dynamic programming (DP) (Bellman, 1953; Cormen et al., 2001) is a mathematical and computer algorithmic scheme for solving optimization problems. The method builds the final solution by expanding initial conditions step by step into more complex cases with $cr(teff)$ determined by the number of states and the complexity of the step.

By means of DP it is possible to obtain a complete solution with $cr(gopt) = true$ in polynomial time complexity by (i) calculating an initial solution for $h = 1$ and pu_1^1 and (ii) expanding the solution upon all of the *pu* and *year*. Unfortunately, the complexity of our problem generates a state-space too large for any direct approach. However, a combined approach of DP and local search can be derived if only we are able to separate a simple subproblems for DP.

We have used a DP approach to determine, for a fixed hour h, the costs of volumes $vol_h^s \in \sum_i v(pu_i^s, h)$ for all $ps_s \in \{ps_s\}$ and generate the maximized total profit pr_h for hour h. In the basic version we have determined pr_h by managing costs of production of each pu_i^s. These costs included the joint costs of maintaining the pu_i^s on $vol_{i,h}^s$ along with a few other parameters (e.g. profit and cost associated with be_b production). In the first step of the algorithm we have generated a lookup table of production volumes vol_h^s and minimal costs of achieving them for each ps_s. Assume that, for a fixed h and ps_s, $v(pu_i^s, h) \in [mn_i, mx_i]$. We denote the cost of pu_i^s in h with volume $x \in [mn_i, mx_i]$ as $ep(i,x)$. Let $m[x][z]$, where $x \in [1, U_s]$ and $z \in [\sum_{i=1}^{U_s} mn_i, \sum_{i=1}^{U_s} mx_i]$ be an optimal cost of the production units $(pu_1^s - pu_x^s)$ generating a total production equal to z. The values of $m[x][z]$ are equal to: (i) $cp(1,z)$ for $z \in [mn_1, mx_1]$, (ii) $\min(m[x-1][z-i] + cp(x,i))$ for $i \in [mn_i, mx_i]$ or (iii) ∞ in all remaining cases. This relation gave us, for each h and ps_s, an optimal configuration of $vol_{i,h}^s$ necessary to produce vol_h^s. In the second step we merged all obtained lookup tables (using DP) along with purchases ze_i modeled as an artificial *pu* with its own lookup table. In the following step we generated (using DP once again) a lookup table for hour h and for sales se_i, containing the optimal methods of selling of particular *en* volumes. In the last step of the algorithm we have compared the two obtained lookup tables and greedily chosen the best vol_h as the production volume of the *ec*.

The basic DP algorithm described above was fast ($cr(teff) = 5 - 8ms$) comparing to CP but did not include $cr(long)$ leading to a very complicated local search with poor final result for the year. Thus, we have refined the solution by introducing partial op-

timization of goods to the DP algorithm. The best configuration we have obtained by introducing only one hard constraint - CO2 emission permit, leading to $cr(teff) = 50ms$.

The main advantages of this approach are: (i) fast and always optimal solution for a fixed h and (ii) inclusion of $cr(nlin)$ without any additional time efforts. Unfortunately, the overall $cr(long)$ management is poor leading to very complex local search that need to be applied as a superior algorithm.

3.3 Linear and Integer Programming

Linear programming (LP) (Dantzig, 1963) is a natural approach to solving linear problems but does not apply directly to nonlinear problems. This drawback can be partially overcome by including relaxations of nonlinearities in the problem model, it however comes at a cost of loosing accuracy and efficiency. We have tested the LP model with a number of relaxations. First of all it has to be noted that the relaxation of (A.19) is required for the model to be of any use because it relates the production to sales and is required in the balance constraints (A.10). For any piecewise linear function fp, defined as (A.23) a convex hull relaxation (Hooker, 2006) has been used (A.24). In the initial solution of the LP model with (A.24), (A.10) are not satisfied because they contain biases caused by the relaxation. To eliminate the bias the LP model is solved again with additional constraints (A.25) that linearize the $fpv_{s,h}$ around the *steady states* obtained from the first solution. To tighten the relaxation, models of further nonlinearities can be included: startup relaxation (A.26) of (A.16), startup cost relaxation (A.27), (A.28) of (A.21) (where *vstc* is the total cost of startups) , minimum pu_i^s enabled relaxation (A.29) of (A.17). Mixed integer/linear programming (MILP) approach, can also be used to tighten the relaxation by introducing integrality constraints $integer(ar_i)$, $integer(on_{s,i,h})$.

We have tested the LP and MILP approaches, for $inst(ec^r, H^r)$ using the COIN-OR (IBM, 2010) CLP and CBC solvers. The LP model with (A.24) and (A.25) had produced the solution with $cr(teff) = 10ms$, LP model with (A.24), (A.25) and (A.29) had $cr(teff) = 40ms$, the model with additional relaxations (A.27) and (A.28) produced a solution after $cr(st) = 1853s$ with the *timeout* and $cr(teff) = 211ms$. A model with integrality constraints did not produce a solution within 1h.

To summarize, the LP relaxation approach has a number of advantages: taking into account all the long period constraints $cr(long) = true$, producing an optimal solution for the defined model $cr(gopt) = true$, a relatively short $cr(teff) = 10ms$ (LP model with (A.24) even for $inst(ec^r, H^r)$. The main disadvantage is that nonlinearities are accounted for in a very limited scope. Therefore, an additional optimization step has to be used to fully satisfy the nonlinearities.

3.4 Local Search

Local search (LS) (Aarts and Lenstra, 1997) is a meta-heuristic for solving computationally hard optimization problems. For the case of *ec* optimization the LS algorithm is organized as follows: it assumes an initial solution (step 1), repairs the violated nonlinear constraints (step 2) by applying a set of *repair_heuristics*, looks for a better solution (step 3) using a set of *improvement_heuristics* (step 3a). All value assignments $CV = vals(heur, CV)$ (step 2a,3a), are performed by choosing a neighborhood pu_j^k and applying new values to CV such that all linear constraints are satisfied and that only the variables $\{v(pu_j^k, h) : h \in [1, H]\}$ in neighborhood pu_j^k are changed from all production variables $\{v(pu_i^s, h)\}$. LS performs backtracking (*backtrack*) in the cases when constraints cannot be repaired to undo wrong heuristic choices. A solution is returned in the form of variable values $values(solution)$ and the corresponding $profit$. The details of particular heuristics shall not be discussed because they are dependent on a particular class of problem instances and are subject to customization.

1. $CV = values(init)$, $profit = -\infty$
2. for all $heur \in repair\_heuristics$:
   (a) for all $cstr \in violated(heur, CV)$: $CV = values(heur(cstr), CV)$
   (b) if $\neg \#violated(heur, CV) = 0$ then *backtrack*
3. for all $heur \in improvement\_heuristics$:
   (a) $CV = values(heur, CV)$
   (b) if $\#violated(all, CV) = 0 \wedge \omega(CV) > profit$ then $values(solution) = CV, profit = \omega(CV)$ else *backtrack*

The advantages of local search is that it can be applied to large and nonlinear problems. The disadvantage is that any LS algorithm is custom tailored for a specific problem definition and class of problem instances. It is typical that LS is highly dependent on the quality of the initial solution $CV = values(init)$ (how many constraints, and of which classes, are violated, are those difficult for LS to handle satisfied etc.)

Table 1: Execution times of the hybrid solver (CP+LP+LS) for several problem instances derived from $inst(ec^r, H^r)$.

Nr	Problem instance ($inst(ec,H)$)	CP+LP	CP+LP+LS
1	Original problem instance for a year ($inst(ec^r, H^r)$)	46.8s	161.8s
2	3 quarters of the year ($inst(ec^{r1}, H^r \cdot 3/4)$)	31.3s	111.1s
3	2 quarters of the year ($inst(ec^{r2}, H^r \cdot 2/4)$)	19.8s	74.2s
4	1 quarters of the year ($inst(ec^{r3}, H^r \cdot 1/4)$)	9.4s	41.3s
5	Higher costs of energy production ($inst(ec^{r4}, H^r)$)	12.7s	121.9s
6	Unconstrained sales and purchases ($inst(ec^{r5}, H^r)$)	183.9s	295.5s
7	Unconstrained sales and purchases, higher energy production cost ($inst(ec^{r6}, H^r)$)	448.2s	614.2s
8	Unconstrained sales and purchases, heavier constrained production gradient ($inst(ec^{r7}, H^r)$)	372.4s	490.7s

3.5 Hybrid Approach

Due to the fact that a single method approach is insufficient to time-efficiently account for all the components of the *ec* profit optimization problem a hybrid approach (solver) has been developed comprising of CP+LP+LS. The hybrid solver takes advantage of the interactions and key strengths of the methods included.

The function of CP has been reduced to performing bound propagation (Dechter and van Beek, 1997) on the constraints (A.8)-(A.14)(A.19) in order to determine *correction variables*[1] for the equality constraints in the LP model and determine infeasible constraints at the outset without time consuming proof of infeasibility by the LS.

The LP model with (A.24) and (A.25) has been used as described in Section 3.3 to obtain an initial solution for the LS. The LP model is preferred to DP because it accounts for all long period constraints (A.12)-(A.14), these constraints are "difficult" to satisfy by the LS (which leads to LS timeouts) if they are not accounted for in the initial solution. Secondly the LP model outperforms DP with respect to $cr(teff)$. The final optimization stage is LS which uses the solution provided by the the LP model as the initial variable assignment. LS is meant to account for the nonlinearities, find a feasible solution (repair phase) and optimize it (improvement phase).

The hybrid approach is supreme because it is the only one that accounts for the complete problem model and produces an acceptable solution for $inst(ec^r, H^r)$ without a timeout. The hybrid solver was tested for several problem instances, derived by modification from the industrial real life problem instance, and the results are presented in Table 1. The results show that the solver execution time is linearly dependent on the size of the problem (time horizon modifications in row 1 to 4) and is strongly dependent on the problem structure i.e. the types of constraints (tightened or relaxed) and profit ratios for control variables. This vulnerability to modification of problem structure is a feature of LP.

[1] correction variables are used to compensate rounding errors performed by the LP solver

4 CONCLUSIONS AND RELATED WORK

Hybridization is an approach to optimization problems that often yields shorter computation times than single method approaches. The relative advantage of hybrid solvers can range up to a few orders of magnitude. This means that for applications with timeouts imposed on optimization it may be the only applicable solution. Furthermore, real life problems often contain heterogeneous constraints and hybridization allows to choose techniques best suited for particular classes of constraints and let them exchange information. A survey of computational results performed by John N.Hooker in (Hooker, 2006) lists some applications of hybrid solvers and their advantages over single method approaches to problem such as: "Scheduling with earliness and tardiness cost" (Beck and Refalo, 2003) solved 5 times more problem instances, "Polyprophylene batch scheduling" (Timpe, 2002) solved previously insoluble problem in 10min, "Lesson timetabling" (Focacci et al., 1999) 2 to 50 times faster, "Min-cost multiple machine scheduling" (Jain and Grossmann, 2001) 20 to 2000 times faster, "Product configuration" (Thorsteinsson and Ottosson, 2002) 30 to 40 times faster.

For the *ec* optimization problem, on the basis of our experiments (Section 3 and Table 2), we believe that a hybrid approach (Section 3.5 and Table 2 entry 7) is the only one that can achieve performance sufficient to meet the requirements of an industrial appli-

Table 2: Comparison of experimental results for different approaches to $inst(ec^r, H^r)$.

Nr	Method	Comment	$cr(gopt)$	$cr(nlin)$	$cr(long)$	$\neg timeout$	$cr(teff)$
1	C(L)P	3 approaches tested	*true*	*true*	*true*	*false*	*na*
2	DP	basic algorithm	*true*	*true*	*false*	*true*	8*ms*
3	DP	with CO2 constraints	*true*	*true*	*false**	*true*	50*ms*
4	LP	with (A.24)(A.25)	*true*	*false**	*true*	*true*	10*ms*
5	LP	with (A.24)(A.25)(A.29)	*true*	*false**	*true*	*true*	40*ms*
6	DP + LS		*false*	*true*	*true*	*false*	*na*
7	CP+LP+LS		*false*	*true*	*true*	*true*	58*ms*

$*$ partially taken into account.

cation.

We have presented a generalized definition of the *ec* optimization problem. We have also laid out the overall structure and details of a hybrid solver developed for the generalized problem, indicating a promising area of research and leaving room for customization (especially in the LS area). We have also discussed approaches which have been discarded at an early stage of development because of their low performance, indicating areas of development which are unlikely to yield satisfactory results. To the best knowledge of the authors no other solution to the *ec* optimization problem has been reported.

The presented hybrid CP+LP+LS approach is generalizable because many complex optimization problems (other than *ec* optimization) can also be decomposed into linear and nonlinear components and then subjected to CP bound consistency, solved by LP to produce an initial solution that is next extended to a feasible solution with respect to nonlinearities by LS and improved by LS, in the same manner as described in this paper.

REFERENCES

Aarts, E. and Lenstra, J., editors (1997). *Local Search in Combinatorial Optimization*. Discrete Mathematics and Optimization. Wiley, Chichester, UK.

Apt, K. (2009). *Principles of Constraint Programming*. Cambridge University Press, New York, NY, USA, 1st edition.

Beck, J. C. and Refalo, P. (2003). A hybrid approach to scheduling with earliness and tardiness costs. *Annals of Operations Research*, 118:49–71. 10.1023/A:1021849405707.

Bellman, R. (1953). An introduction to the theory of dynamin programming. *The RAND Corporation*, Report R-245.

Cisco Systems (2010). ECLiPSe constraint logic programming system. http://www.eclipse-clp.org/.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to algorithms (2nd edition)*. MIT Press and McGraw-Hill.

Dantzig, G. B. (1963). *Linear programming and extensions*. Princeton Univ. Press, Princeton, NJ.

Dechter, R. and van Beek, P. (1997). Local and global relational consistency. *Theor. Comput. Sci.*, 173:283–308.

Focacci, F., Lodi, A., and Milano, M. (1999). Cost-based domain filtering. In *Proceedings of the 5th International Conference on Principles and Practice of Constraint Programming*, CP '99, pages 189–203, London, UK. Springer-Verlag.

Hooker, J. N. (2006). *Operations research methods in constraint programming*, pages 525–568. Handbook of Constraint Programming. Elsevier, Amsterdam.

IBM (2010). Coin-or computational infrastructure for operations research. http://www.coin-or.org/ .

Jain, V. and Grossmann, I. E. (2001). Algorithms for hybrid milp/cp models for a class of optimization problems. *INFORMS J. on Computing*, 13:258–276.

Marriott, K. and Stuckey, P. J. (1998). *Introduction to Constraint Logic Programming*. MIT Press, Cambridge, MA, USA.

Schulte, C. (2010). Gecode constraint programming system. http://www.gecode.org/.

Szymanek, R. and Kuchciński, K. (2010). Jacop constraint programming system. http://jacop.osolpro.com/.

Thorsteinsson, E. S. and Ottosson, G. (2002). Linear relaxations and reduced-cost based propagation of continuous variable subscripts. *Annals of Operations Research*, 115:15–29. 10.1023/A:1021136801775.

Timpe, C. (2002). Solving planning and scheduling problems with combined integer and constraint programming. *OR Spectrum*, 24:431–448. 10.1007/s00291-002-0107-1.

APPENDIX A

The appendix contains equations of the problem model and its relaxation. **APPENDIX A** is available at http://sun.aei.polsl.pl/∼twojdyla/opt/ec_opt_appA.pdf.