Scheduling Home Hospice Care with Logic-Based Benders Decomposition

Aliza Heching[1] and J. N. Hooker[2]

[1] Compassionate Care Hospice Group, aliza.heching@cchnet.net
[2] Carnegie Mellon University, Pittsburgh, USA, jh38@andrew.cmu.edu

Abstract. We propose an exact optimization method for home hospice care staffing and scheduling, using logic-based Benders decomposition (LBBD). The objective is to match hospice care aides with patients and schedule visits to patient homes, so as to maximize the number of patients serviced by available staff, while meeting requirements of the patient plan of care and scheduling constraints imposed by the patients and the staff. The Benders master problem assigns aides to patients and days of the week and is solved by mixed integer programming (MIP). The routing and scheduling subproblem decouples by aide and day of the week and is solved by constraint programming. We report preliminary computational results for problem instances obtained from a major hospice care provider. We find that LBBD is superior to state-of-the-art MIP and solves problems of realistic size, if the aim is to conduct staff planning on a rolling basis while maintaining continuity of the care arrangement for patients currently receiving service.

Keywords: home health care problem, routing and scheduling, logic-based Benders decomposition, home hospice care

1 Introduction

Home health care is one of the world's most rapidly growing industries, due primarily to cost advantages as well as aging populations. Home care allows patients to receive basic medical or hospice care in comfortable and familiar surroundings, rather than being transported or admitted to facilities that are expensive to operate. It also reduces the risk of acquiring drug-resistant infections that may spread in hospitals and nursing homes. The increasing availability of portable equipment and online consultation makes home care feasible for an ever wider variety of conditions.

The cost-effectiveness of home health care depends critically on the efficient dispatch of health care aides, whom we call *aides* for short. This poses the *home health care problem* (HHCP), which asks how home visits can be scheduled and staffed so as to make the best use of aides while meeting patient needs. Aides typically start their work shift at home or a central office, travel directly from one patient to the next, and return to home or office at the end of the shift. The shift may be subject to a number of legal or contractual restrictions, such as a maximum work time and the need for lunch/dinner breaks. Each medical or hospice service must be performed by an aide with the proper qualifications, and services may be restricted to specified days or time windows. It may be necessary for two or more aides to visit a patient at the same time, to carry out more complicated treatments.

We focus on hospice care, which has a few distinctive characteristics. Aides frequently provide personal and household services rather than medical treatment, or they may simply offer companionship. They tend to visit on a regular schedule over a period of several weeks, such as three times a week in the morning. It is often important for a given service to be provided by the same aide during every visit, so far as is possible. Staff planning is typically over a longer time horizon, perhaps several weeks.

Because of the regularity of visits and the need for staffing continuity, the primary challenge that arises in practice is to update the schedule and anticipate staffing needs as the patient population evolves. If patient turnover for the next few weeks can be forecast, then a schedule can be computed for the new population to determine what kind of work force will be required.

We therefore address the problem of recomputing the staff assignments and visitation schedule when a specified subset of the patients are replaced by new patients with known requirements. Due to the importance of continuity, we require that existing patients be served by the same aide on the same days as before, but allow for adjustments in the time of day. The models are easily modified to maintain the time of day as well, or to reschedule both the time of day and days of the week.

Due to the difficulty of the HHCP, nearly all existing solution methods are heuristic algorithms. Recent work can be found in [1–10]. The few exact methods include two branch-and-price methods [11, 12] and a branch-and-bound method that relies on a traveling salesman algorithm [13].

We propose a very different exact method that uses logic-based Benders decomposition (LBBD) [14–18] and is well suited to scheduling on a rolling basis. An exact method offers the advantage that one can know with certainty whether a given work force can cover anticipated patient needs, and therefore when hiring additional staff is really necessary. We find that LBBD makes exact solution possible for applications of realistic size when the problem is to reschedule on a rolling basis, rather than schedule all the patients from scratch.

LBBD exploits a natural decomposition of the HHCP into an assignment component (allocation of patients to aides) and a routing and scheduling component (dispatching and routing of aides). It combines the complementary strengths of mixed integer programming (MIP) and constraint programming (CP), with MILP solving the assignment problem and CP solving the routing and scheduling problem.

LBBD is a generalization of classical Benders decomposition [19] in which the subproblem can be any combinatorial problem, not necessarily a linear programming problem. The Benders cuts are based on an inference dual of the subproblem, whose solution is regarded as a proof of optimality or infeasibility, rather than a linear programming dual. LBBD has reduced solution times by orders of magnitude relative to conventional methods in a variety of problems [14, 15, 17, 18, 20–22, 16, 23–34]. In our solution of the HHCP, the Benders master problem assigns aides to patients and to days of the week on which these patients are serviced. The Benders subproblem is the routing and scheduling problem that results from the assignment obtained by solving the master problem. The subproblem decouples into routing and scheduling micro-problems that correspond to each aide and each day of the week. Infeasible micro-problems give rise to Benders cuts that are added to the master problem. The process repeats until all the micro-problems are feasible. Our primary methodological contribution is to identify a relaxation of the scheduling subproblem that, when included in the master problem, results in significantly faster solution.

The only previous application of LBBD to the HHCP of which we are aware is a heuristic method in an unpublished manuscript [4]. It solves the master problem with greedy heuristic and the subproblem with CP, while creating a schedule for only one day.

2 The Problem

The problem can be stated as follows. For each patient j there is a time window $[r_j, d_j]$ during which a visit to that patient must take place, as well as the visit duration p_j. It is assumed that each patient requires one type of visit. If a patient requires two or more types of visits, the patient is regarded as two or more distinct patients (with nonoverlapping time windows if the visits should not overlap). Aides must be qualified to serve assigned patients, but this requirement actually makes the problem easier to solve and is therefore not considered here.

Each aide i departs from home base b_i and returns to home base b'_i (which could be the same as b_i). The allowable shift hours of aide i are specified by a time window $[r_{b_i}, d_{b_i}]$ for departure from the origin base and a window $[r_{b'_i}, d_{b'_i}]$ for arrival at the destination base. Travel time between patient (or home base) j and patient j' is $t_{jj'}$.

We formulate the problem for a cyclic 7-day schedule with no visits on weekends. Each patient j requires v_j visits per week, with $v_j \in \{1, 2, 3, 5\}$. Twice-a-week visits must be separated by at least 2 days, and thrice-a-week visits by 1 day. The variables are designed to facilitate a decomposition scheme in which the scheduling subproblem is solved by constraint programming. Binary variable $\delta_j = 1$ when patient j is serviced, and binary variable $x_{ij} = 1$ when aide i is assigned to patient j. Binary variable $y_{ijk} = 1$ when aide i visits patient j on day k, so that $y_{ijk} \leq x_{ij}$ for all i, j, k. There are sequencing variables $\pi_{ik\nu}$ that represent the νth patient visited by aide i on day k. Variable s_{ijk} indicates the time at which aide i's visit to patient j starts on day k.

We maximize the number of patients that can be covered by a given work force. This not only determines whether the work force is adequate, but it tends to minimize idle time and driving time in an aide's schedule. The problem can be stated as follows:

$$
\begin{array}{lll}
\max \displaystyle\sum_j \delta_j & (a) & \\
\displaystyle\sum_i x_{ij} = \delta_j, \;\; \sum_{i,k} y_{ijk} = v_j \delta_j, \;\; \text{all } j & (b) & \\
y_{ijk} \leq x_{ij}, \;\; \text{all } i, j, k & (c) & \\
y_{ib_i k} = y_{ib'_i k} = 1, \;\; \text{all } i, k & (d) & \\
y_{ij,k+\tau} \leq 1 - y_{ijk}, \; \tau = 1, 4 - v_j, & & \\
\qquad\qquad \text{all } i, j, k \text{ with } v_j \in \{2,3\}, \, 1 \leq k \leq v_j + 1 & (e) & (1) \\
\delta_j, x_{ij}, y_{ijk} \in \{0,1\}, \;\; \text{all } i, j, k, & (f) & \\
n_{ik} = \displaystyle\sum_j y_{ijk}, \;\; \text{alldiff}\big\{\pi_{ik\nu} \;\big|\; \nu = 1, \ldots, n_{ik}\big\}, \;\; \text{all } i, k & (g) & \\
\pi_{ik\nu} \in \{j \mid y_{ijk} = 1\}, \;\; \text{all } i, k, \text{ and } \nu = 1, \ldots, n_{ik} & (h) & \\
\pi_{i1k} = b_i, \; \pi_{in_{ik}k} = b'_i, \;\; \text{all } i, k & (i) & \\
r_j \leq s_{ijk} \leq d_j - p_j, \;\; \text{all } i, j, k & (j) & \\
s_{\pi_{ik\nu}} + p_{\pi_{ik\nu}} + t_{\pi_{k\nu}\pi_{k,\nu+1}} \leq s_{\pi_{ik,\nu+1}}, \;\; \text{all } i, k, \text{ and } \nu = 1, \ldots, n_{ik} - 1 & (k) &
\end{array}
$$

Constraint (b) defines δ_j and ensures that every patient is visited by the same aide on the required number of days. Constraint (d) says that an aide's start and end home base must be visited every day. Constraint (e) controls the spacing of assigned days. Constraint (g) defines variable n_{ik} to be the number of patients assigned to aide i on day k and requires that the corresponding sequence variables take distinct values. Constraint (h) says that an aide's visits that are sequenced on a given day are in fact those assigned to the aide on that day. Constraint (i) ensures that the start and end home base are visited first and last, respectively. Constraint (j) enforces time windows. Constraint (k) ensures that a visit does not start before the aide can arrive from the previous visit.

When updating an existing schedule, we need only fix $y_{ijk} = 1$ when patient j remains in the population and is assigned to aide i on day k. To require that existing patients be served at the same time of day as before, their time windows can be set equal to the visit period. To allow existing patients to be served on different days of the week than before, we can fix the variables x_{ij} rather than y_{ijk}.

3 Benders Subproblem

The subproblem decouples into a separate micro-problem for each aide and each day. Each is a feasibility problem that checks whether there is a schedule that observes the time windows while taking account of the visit durations and travel times. If not, a Benders cut is generated as described below.

The subproblem formulation consists of the scheduling constraints in (1) after the daily assignment variables y_{ijk} are fixed to the values $\bar{y}_{ijk}$ they receive in the previous solution of the master problem. The micro-problem S_{ik} for each aide i and day k is

$$\begin{aligned}
&\text{alldiff}\{\pi_\nu \mid \nu = 1,\ldots,\bar{n}_{ik}\} \\
&\pi_1 = b_i,\ \pi_{\bar{n}_{ik}} = b'_i \\
&r_j \le s_j \le d_j - p_j,\ \ \text{all } j \in P_{ik} \\
&s_{\pi_\nu} + p_{\pi_\nu} + t_{\pi_\nu \pi_{\nu+1}} \le s_{\pi_{\nu+1}},\ \ \nu = 1,\ldots,\bar{n}_{ik} - 1 \\
&\pi_\nu \in P_{ik},\ \ \nu = 1,\ldots,\bar{n}_{ik}
\end{aligned}$$

where $P_{ik} = \{j \mid \bar{y}_{ijk} = 1\}$ and $\bar{n}_{ik} = |P_{ik}|$. If S_{ik} is infeasible, we generate a simple nogood cut $\sum_{j\in P_{ik}}(1 - y_{ijk}) \ge 1$ that prevents the same set of patients from being assigned to aide i on day k in subsequent assignments.

We can, in principle, generate stronger cuts by determining whether the same proof of infeasibility remains valid when smaller sets of patients are assigned to aide i on day k. Unfortunately, we do not have access to the mechanism by which CP solver proves infeasibility. We therefore tease out stronger cuts by re-solving S_{ik} for subsets of P_{ik}. S_{ik} can be rapidly re-solved because of its small size. We use the following simple heuristic, which has proved effective in several studies [18, 20, 22, 21, 34]. We initially set $\bar{P}_{ik} = P_{ik}$, and for each $j \in \bar{P}_{ik}$ we do the following: remove j from $\bar{P}_{ik}$, re-solve S_{ik}, and restore j to $\bar{P}_{ik}$ if the modified S_{ik} is feasible. This yields a Benders cut that results in significantly better performance :

$$\sum_{j\in\bar{P}_{ik}} (1 - y_{ijk}) \ge 1 \tag{2}$$

Whenever we derive a cut for a given aide i and day k of the week, we can generate a similar cut for every other day of the week. However, the resulting proliferation of cuts causes the solution of master problem to bog down. We found that an effective compromise is to sum the cuts for the remaining 4 weekdays. Thus for each cut (2), we also generate the cut

$$\sum_{k'\neq k} \sum_{j\in\bar{P}_{ik}} (1 - y_{ijk'}) \ge 4$$

4 Benders Master Problem

The basic master problem consists of constraints (a)–(f) of the original problem (1) and the Benders cuts generated in all previous iterations as described above. It also contains a relaxation of the subproblem, because computational experience in [22] and elsewhere indicates that including such a relaxation is crucial to obtaining good performance of LBBD.

We found the following *time window relaxation* to be effective. For each aide i, define a set $\{[r_{b_i}, \alpha_{i\ell}] \mid \ell \in L_i\}$ of *backward intervals* that begin with the start of the aide's shift, and a set $\{[\beta_{i\ell}, d_{b'_i}] \mid \ell \in L'_i\}$ of *forward* intervals that end with the termination of the shift. For each backward interval $\ell \in L_i$, let $J_{i\ell}$ be the set of visits whose time window $[r_j, d_j]$ is a subset of the interval, and define $J'_{i\ell}$

similarly for forward intervals. Let the *backward augmented duration* p'_{ijk} for a visit j, aide i and day k be the duration p_j plus the minimum transit time from the previous visit (which may be the origin base for the aide), and similarly for the *forward augmented duration* p''_{ijk}. That is,

$$p'_{ijk} = p_j + \min\big\{t_{b_i j},\ \min_{j' \in Q_{ik}}\{t_{j'j}\}\big\},\ \ p''_{ijk} = p_j + \min\big\{\min_{j' \in Q_{ik}}\{t_{jj'}\}, t_{jb'_i}\big\}$$

where Q_{ik} is the set of visits that are already assigned aide i on day k, or that have not yet been assigned an aide. Thus the backward augmented duration is a lower bound on the time required to reach and carry out a visit, and similarly for the forward augmented duration.

We now observe that sum of the backward augmented durations of visits in $J_{i\ell}$ must be at most the width of backward interval ℓ, and similar for any forward interval:

$$\sum_{j \in J_{i\ell}} p'_{ijk} y_{ijk} \le \alpha_{i\ell} - r_{b_i},\ \ell \in L_i;\quad \sum_{j \in J'_{i\ell}} p''_{ijk} y_{ijk} \le d_{b'_i} - \beta_{i\ell},\ \ell \in L'_i \tag{3}$$

This because the visits and travel to each visit must fit between the beginning of the aide's shift and the end of the backward interval, and similarly for a forward interval. Inequalities (3), collected over all aides i, comprise a time window relaxation.

The backward and forward intervals should be chosen so that the visits that can take place within them have a large total duration relative to the width of the interval, as this results in tighter inequalities (3). In the test instances, the time windows of the visits span either most of the morning or most of the afternoon. It was therefore natural to use one backward interval ending at noon, and one forward interval beginning at noon, for each aide i. Thus $L_i = L'_i = \{1\}$ and $\alpha_{i1} = \beta_{i1} =$ noon for each i.

This is a weak relaxation when scheduling all patients from scratch, because the shortest travel time from the last (or next) visit is a weak bound on the actual travel time. However, it is more effective in the rolling problem, because the shortest travel time is computed only over patients who are already assigned aide i on day k or are unassigned.

5 Computational Results

We tested the LBBD algorithm on real-world data provided by a major hospice care firm. To obtain an initial schedule, we ran a greedy heuristic on an 80-patient population using 20 aides. Since the heuristic could schedule only 48 patients, we ran the LBBD algorithm on 60 of these patients, including 40 pre-scheduled by the greedy heuristic and 20 treated as new patients. LBBD scheduled all of the new patients using 18 aides. The resulting 60-patient schedule was used as a starting point for computational tests. It is better than a heuristic schedule but worse than an optimal one, as one might expect when scheduling on a rolling basis.

We compared the performance of LBBD and mixed integer programming (MIP) for different rates of patient turnover in the 60-patient population. One instance is generated for each number $m = 6, \ldots, 23$ of new patients, where the new patients are assumed to be the last m patients in the list of 60. We designated 8 of the 18 aides as available to cover the new patients (along with their pre-assigned patients), because a minimum of 9 aides were required in nearly every instance. This allowed us to test computational performance near the phase transition for the problem.

We formulated an MIP model for the problem by modifying the well-known multicommodity flow model for the vehicle routing problem with time windows [35–37]. The model consists of (a)–(f) in (1) and the following:

$$\begin{aligned}
& w_{ijb'_i k} + \sum_{j' \neq j} w_{ijj'k} = w_{ib_i jd} + \sum_{j' \neq j} w_{ij'jk} = y_{ijk}, \ \text{ all } i, j, k \\
& w_{ib_i jk} + \sum_{j' \neq j} w_{ijj'k} = w_{ijb'_i k} + \sum_{j' \neq j} w_{ijj'k}, \ \text{ all } i, j, k \\
& s_{ij'k} \geq s_{ijk} + p_j + t_{jj'} - M_{jj'}(1 - w_{ijj'k}), \ \text{ all } i, j, j', k \\
& r_{b_i} \leq s_{ib_i k} \leq d_{b'_i}, \ \ r_j \leq s_{ijk} \leq d_j - p_j, \ \text{ all } i, j, k
\end{aligned}$$

plus similar constraints in which j and/or j' is a home base. Here the binary variable $w_{ijj'k} \in \{0, 1\}$ represents flow and $M_{jj'} = \max\{0,\ d_j - p_j + t_{jj'} - r_{j'}\}$.

We implemented LBBD using the IBM ILOG CPLEX Optimization Studio version 12.6.2. The master problem was solved by CPLEX and the subproblem by the IBM ILOG CP Optimizer. The routing and scheduling micro-problems were formulated with a noOverlap constraint associated with sequencing and interval variables. We solved the MIP model using CPLEX. The CPLEX presolve routine removes variables in the MIP model and LBBD master problem that are fixed to 0 or 1 by preassignments. The solver was run in Windows 7 on a laptop with an Intel Core i7 processor and 7.75 GB RAM.

The results appear in Table 1. Since ILOG Studio does not report solution time for LBBD, the times shown are total elapsed clock times as indicated on the Studio console. They reflect overhead incurred in setting up the problem and retrieving the solution, which can be a significant fraction of total time for the smallest instances.

Both LBBD and MIP readily solve the smaller instances, but MIP suffers a combinatorial blowup when there are more than 14 or 15 new patients. MIP is disadvantaged by the fact that the number of variables grows quadratically with the number of new patients, while in LBBD it grows only linearly. LBBD therefore postpones the blowup significantly. Table 1 also shows that including a subproblem relaxation in the master problem is crucial to the performance of LBBD.

Patient records suggest that a 5–8% turnover per week is typical in practice. LBBD therefore allows staff planning a month or so in advance for a patient population of 60. This is adequate for many real-world problem instances, particularly given that improvements in the LBBD model and subproblem relaxation are likely.

Table 1. Effect of patient turnover on computation times in a population of 60 patients and 18 aides, 8 of whom are available for new patients. The new patients replace an equal number of existing patients. Number of Benders iterations is shown, along with computation time (minutes : seconds). The last two columns show results for LBBD without a subproblem relaxation in the master problem.

New	Patients	LBBD		MIP	LBBD no relax	
Patients	Scheduled	Iters.	Time	Time	Iters.	Time
6	60	2	**0:10**	0:39	17	1:17
7	60	3	**0:15**	0:39	18	1:23
8	60	7	**0:34**	0:49	22	1:49
9	59	7	**0:34**	0:41	20	1:38
10	59	6	**0:31**	0:43	20	1:41
11	59	6	**0:32**	0:41	31	2:52
12	59	9	**0:47**	0:45	30	2:54
13	59	24	2:15	**1:00**	51	6:53
14	59	29	**3:00**	20:27	63	9:18
15	59	37	**4:20**	11:40	72	11:57
16	59	39	**4:45**	142:08	87	16:26
17	59	39	**4:46**		129	36:39
18	59	38	**4:56**		126	30:00
19	59	75	**14:13**		138	48:01
20	58	75	**14:44**		141	63:49
21	58	87	**24:21**			
22	59	130	**48:00**			
23	59	159	**93:56**			

6 Conclusion

We find that logic-based Benders decomposition solves the home hospice care problem on a rolling basis more rapidly than state-of-the-art mixed integer programming, and it scales up to problems of realistic size. Unlike nearly all competing methods developed for this problem, it computes an optimal schedule and therefore allows planners to determine with certainty whether a given work force can meet projected patient requirements.

LBBD has the advantage that the routing and scheduling subproblems remain constant in size as the patient population grows, while the number of scheduling variables in MIP increases quadratically. The performance of LBBD also benefits from an effective time-window relaxation of the subproblem that we developed for inclusion in the master problem. LBBD is particularly well suited for scheduling on a rolling basis because continuity constraints strengthen this relaxation.

Due to the sensitivity of performance to the quality of the subproblem relaxation, future research will focus on identifying tighter relaxations, as well as incorporating constraints and objectives that more adequately reflect the complexity of the real-world problem.

References

1. Hertz, A., Lahrichi, N.: A patient assignment algorithm for home care service. Journal of the Operational Research Society **60** (2009) 481–495
2. Trautsamwieser, A., Hirsch, P.: Optimization of daily scheduling for home health care services. Journal of Applied Operational Research **3** (2011) 124–136
3. Nickel, S., Schröder, M., Steeg, J.: Mid-term and short-term planning support for home health care services. European Journal of Operational Research **219** (2012) 574–587
4. Ciré, A., Hooker, J.N.: A heuristic logic-based Benders method for the home health care problem. Presented at Matheuristics 2012, Angra dos Reis, Brazil (2012)
5. Rendl, A., Prandtstetter, M., Hiermann, G., Puchinger, J., Raidl, G.: Hybrid heuristics for multimodal homecare scheduling. In Beldiceanu, N., Jussien, N., Pinson, E., eds.: CPAIOR Proceedings. Volume 7298 of Lectures Notes in Computer Science., Nantes, France, Springer Verlag (2012) 339–355
6. Allaoua, H., Borne, S., Létocart, L., Calvo, R.W.: A matheuristic approach for solving a home health care problem. Electronic Notes in Discrete Mathematics **41** (2013) 471–478
7. Cappanera, P., Scutellà, M.G.: Joint assignment, scheduling and routing models to home care optimization: A pattern-based approach. Transportation Science (published online 2014)
8. S. Yalçındağ, Matta, A., Şahin, E., Shanthikumar, J.G.: A two-stage approach for solving assignment and routing problems in home health care services. In Matta, A., Li, J., Sahin, E., Lanzarone, E., Fowler, J., eds.: Proceedings of the International Conference on Health Care Systems Engineering. Volume 61 of Proceedings in Mathematics and Statistics., New York, Springer (2014) 47–59
9. Mankowska, D.S., Meisel, F., Bierwirth, C.: The home health care routing and scheduling problem with interdependent services. Health Care Management Science **17** (2014) 15–30
10. Rest, K.D., Hirsch, P.: Daily scheduling of home health care services using time-dependent public transport. Flexible Services and Manufacturing Journal (published online 2015)
11. Dohn, A., Kolind, E., Clausen, J.: The manpower allocation problem with time windows and job-teaming constraints: A branch-and-price approach. Computers and Operations Research **36** (2009) 1145–1157
12. Rasmussen, M.S., Justesen, T., Dohn, A., Larsen, J.: The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies. European Journal of Operational Research **219** (2012) 598–610
13. Chahed, S., Marcon, E., Sahin, E., Feillet, D., Dallery, Y.: Exploring new operational research opportunities within the home care context: The chemotherapy at home. Health Care Management Science **12** (2009) 179–191
14. Hooker, J.N.: Logic-based Benders decomposition. In: INFORMS National Meeting. (1995)
15. Hooker, J.N., Yan, H.: Logic circuit verification by Benders decomposition. In Saraswat, V., Hentenryck, P.V., eds.: Principles and Practice of Constraint Programming: The Newport Papers, Cambridge, MA, MIT Press (1995) 267–288
16. Hooker, J.N.: Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction. Wiley, New York (2000)
17. Hooker, J.N., Ottosson, G.: Logic-based Benders decomposition. Mathematical Programming **96** (2003) 33–60

18. Hooker, J.N.: Planning and scheduling by logic-based Benders decomposition. Operations Research **55** (2007) 588–602
19. Benders, J.F.: Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik **4** (1962) 238–252
20. Hooker, J.N.: A hybrid method for planning and scheduling. Constraints **10** (2005) 385–401
21. Hooker, J.N.: An integrated method for planning and scheduling to minimize tardiness. Constraints **11** (2006) 139–157
22. Ciré, A., Çoban, E., Hooker, J.N.: Logic-based Benders decomposition for planning and scheduling: A computational analysis. In Barták, R., Salido, M.A., eds.: COPLAS Proceedings. (2015) 21–29
23. Jain, V., Grossmann, I.E.: Algorithms for hybrid MILP/CP models for a class of optimization problems. INFORMS Journal on Computing **13** (2001) 258–276
24. Harjunkoski, I., Grossmann, I.E.: Decomposition techniques for multistage scheduling problems using mixed-integer and constraint programming methods. Computers and Chemical Engineering **26** (2002) 1533–1552
25. Harjunkoski, I., Grossmann, I.E.: A decomposition approach for the scheduling of a steel plant production. Computers and Chemical Engineering **25** (2001) 1647–1660
26. Liu, W., Gu, Z., Xu, J., Wu, X., Ye, Y.: Satisfiability modulo graph theory for task mapping and scheduling on multiprocessor systems. IEEE Transactions on Parallel and Distributed Systems **22** (2011) 1382–1389
27. Lombardi, M., Milano, M., Ruggiero, M., Benini, L.: Stochastic allocation and scheduling for conditional task graphs in multi-processor systems-on-chip. Journal of Scheduling **13** (2010) 315–345
28. Cambazard, H., Hladik, P.E., Déplanche, A.M., Jussien, N., Trinquet, Y.: Decomposition and learning for a hard real time task allocation problem. In Wallace, M., ed.: Principles and Practice of Constraint Programming (CP 2004). Volume 3258 of Lecture Notes in Computer Science., Springer (2004) 153–167
29. Chu, Y., Xia, Q.: Generating Benders cuts for a class of integer programming problems. In Régin, J.C., Rueher, M., eds.: CPAIOR 2004 Proceedings. Volume 3011 of Lecture Notes in Computer Science., Springer (2004) 127–141
30. Maravelias, C.T., Grossmann, I.E.: Using MILP and CP for the scheduling of batch chemical processes. In Régin, J.C., Rueher, M., eds.: CPAIOR 2004 Proceedings. Volume 3011 of Lecture Notes in Computer Science., Springer (2004) 1–20
31. Maravelias, C.T., Grossmann, I.E.: A hybrid MILP/CP decomposition approach for the continuous time scheduling of multipurpose batch plants. Computers and Chemical Engineering **28** (2004) 1921–1949
32. Terekhov, D., Beck, J.C., Brown, K.N.: Solving a stochastic queueing design and control problem with constraint programming. In: Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI 2007). Volume 1., AAAI Press (2007) 261–266
33. Benini, L., Bertozzi, D., Guerri, A., Milano, M.: Allocation and scheduling for MPSoCs via decomposition and no-good generation. In: Principles and Practice of Constraint Programming (CP 2005). Volume 3709 of Lecture Notes in Computer Science., Springer (2005) 107–121
34. Ciré, A.A., Çoban, E., Hooker, J.N.: Mixed integer programming vs logic-based Benders decomposition for planning and scheduling. In Gomes, C., Sellmann, M., eds.: CPAIOR 2013 Proceedings. (2013) 325–331
35. Desrochers, M., Lenstra, J.K., Savelsbergh, M.W.P., Soumis, F.: Vehicle routing with time windows: Optimization and approximation. In Golden, B.L., Assad,

A.A., eds.: Vehicle Routing: Methods and Studies. North-Holland, Amsterdam (1988) 65–84
36. Desrochers, M., Laporte, G.: Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. Operations Research Letters **10** (1991) 27–36
37. Cordeau, J.F., Laporte, G., Savelsbergh, M., Vigo, D.: Vehicle routing. In Barnhart, C., Laporte, G., eds.: Handbook in Operations Research and Management Science. Volume 14. Elsevier (2007) 367–428