

# Scheduling Home Hospice Care with Logic-based Benders Decomposition

Aliza Heching

Compassionate Care Hospice

John Hooker

Carnegie Mellon University

EURO 2016

Poznan, Poland

# Home Health Care

- Home health care problem.
  - Assign **aides** to homebound **patients**.
    - ...subject to constraints on aide qualifications and patient preferences.
    - One patient may require a team of aides.
  - **Route** each aide through assigned patients, observing **time windows**.
    - ...subject to constraints on hours, breaks, etc.

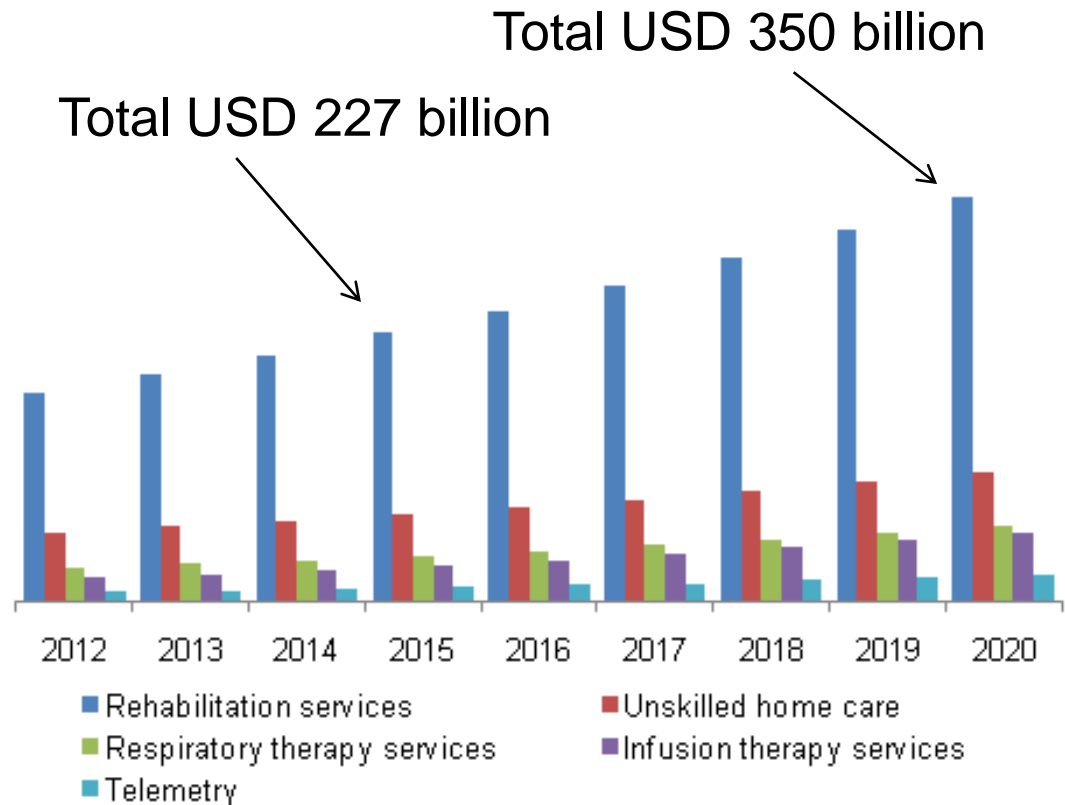


# Home Health Care

- A large industry, and **rapidly growing**.
  - Today, roughly as large as all courier and delivery services.

## Projected Growth Global Home Health Care Industry

By type of service



# Home Health Care

- Advantages of home health care
  - Lower cost
    - Hospital & nursing home care is very expensive.
  - No hospital-acquired infections
    - Less exposure to “superbugs.”
  - Preferred by patients
    - Comfortable, familiar surroundings of home.
    - Sense of control over one’s life.
  - Supported by new equipment & technology
    - IT integration with hospital systems.
    - Online consulting with specialists.

# Home Health Care

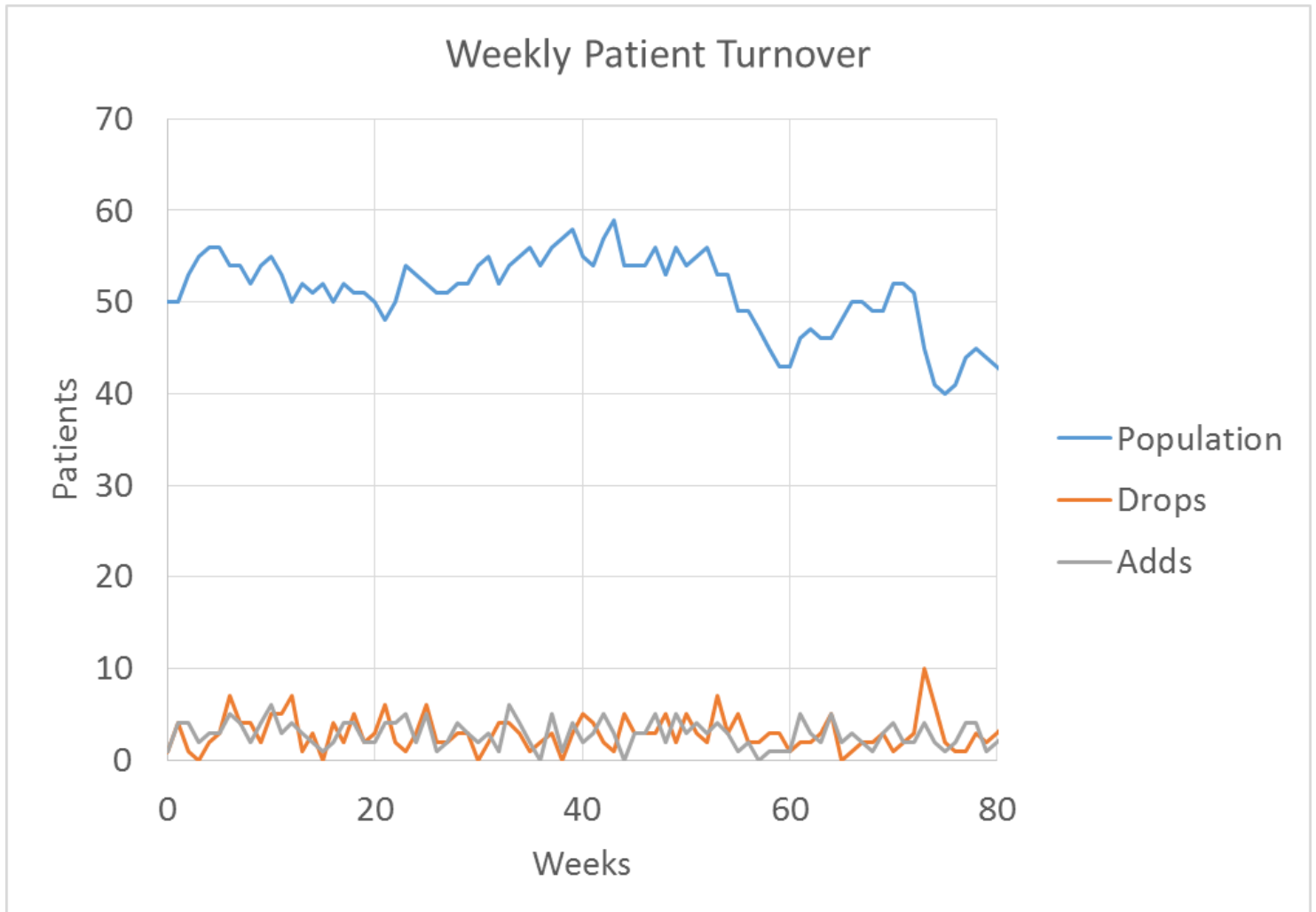
- Critical factor to realize cost savings:
  - Aides must be **efficiently** scheduled.
- This is our task.
  - Focus on home hospice care.



# Home Hospice Care

- Distinguishing characteristics of hospice care
  - Personal & household services
  - Regular weekly schedule
    - For example, Mon-Wed-Fri at 9 am.
  - Same aide each visit
  - Long planning horizon
    - Several weeks
  - Rolling schedule
    - Update schedule as patient population evolves.

# Home Hospice Care



5-8%  
weekly  
turnover

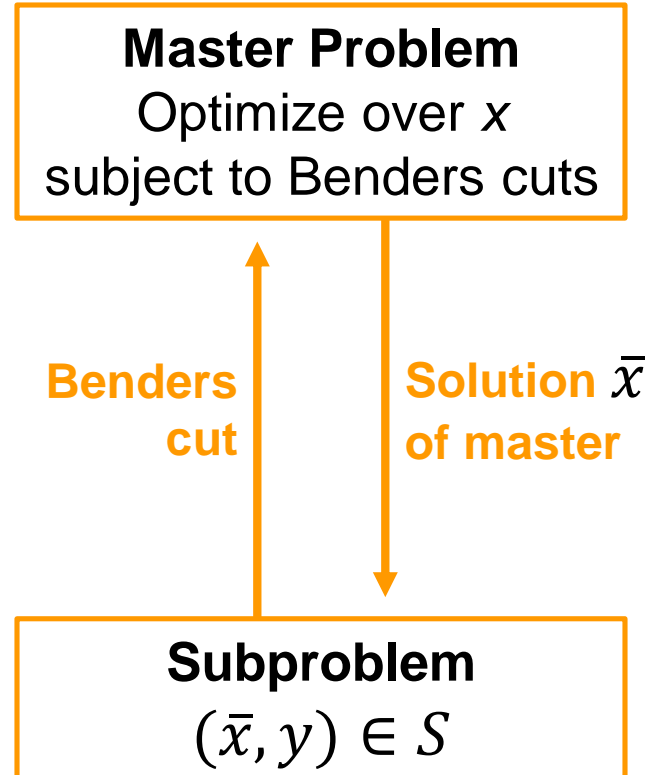
# Logic-based Benders

- **Logic-based Benders decomposition** is a generalization of classical Benders.

- Consider a simplified problem:

$$\begin{aligned} \min f(x) \\ (x, y) \in S \end{aligned}$$

- Benders cut excludes  $\bar{x}$  (and perhaps similar solutions) if it is infeasible in the subproblem.



Benders 1962 (classical method),  
JH 2000, JH & Ottosson 2003 (LBBD)



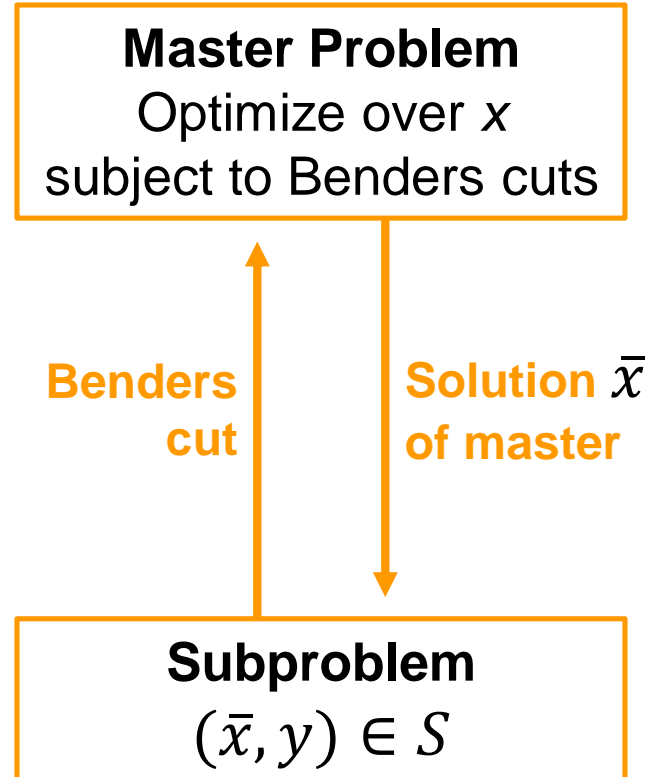
# Logic-based Benders

- **Logic-based Benders decomposition** is a generalization of classical Benders.

- Consider a simplified problem:

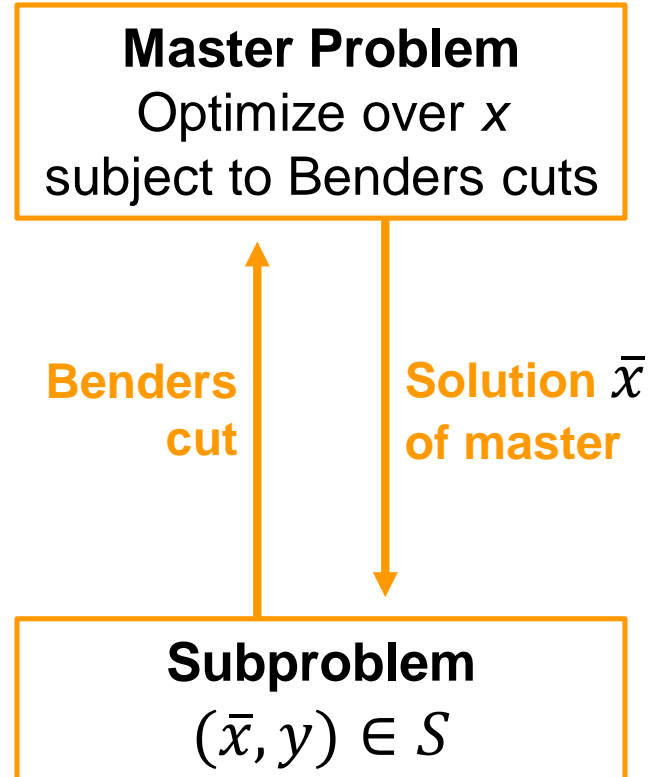
$$\begin{aligned} \min f(x) \\ (x, y) \in S \end{aligned}$$

- Benders cut excludes  $\bar{x}$  (and perhaps similar solutions) if it is infeasible in the subproblem.
- Benders cut based on **inference dual**
- Algorithm terminates when  $\bar{x}$  is feasible in the subproblem.



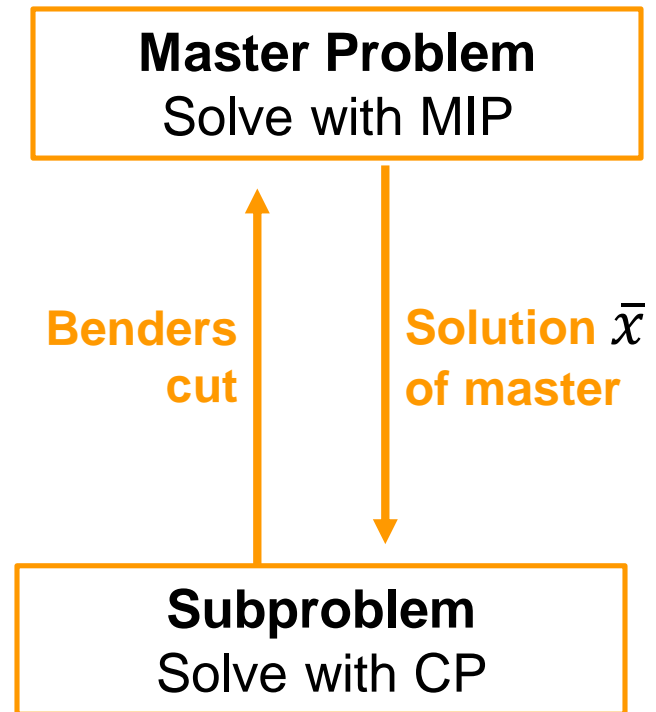
# Logic-based Benders

- **Logic-based Benders decomposition** is a generalization of classical Benders.
  - Master problem contains a **relaxation** of the subproblem.
  - This is critical for good performance.



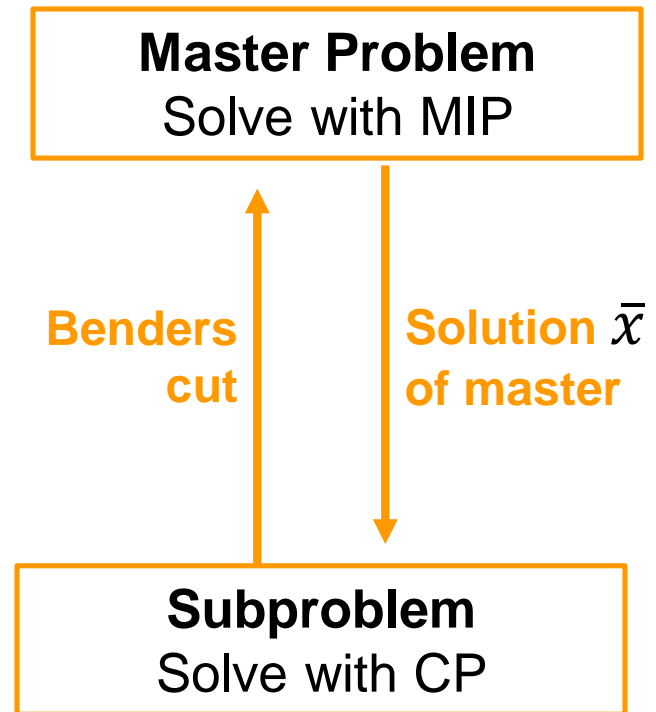
# Home Hospice Care

- Solve with Benders decomposition.
  - **Assign aides to patients** in master problem.
    - Maximize number of patients served by a given set of aides.



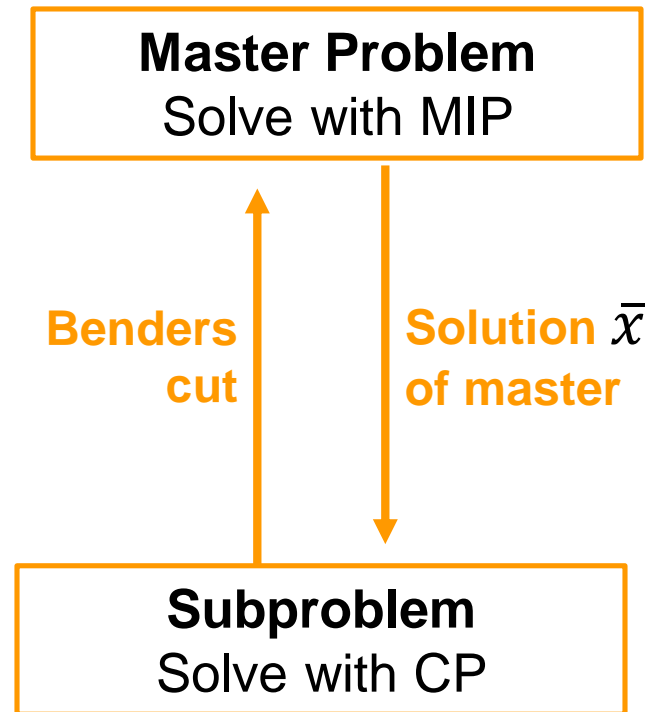
# Home Hospice Care

- Solve with Benders decomposition.
  - **Assign aides to patients** in master problem.
    - Maximize number of patients served by a given set of aides.
  - **Schedule home visits** in subproblem.
    - Cyclic weekly schedule.
    - No visits on weekends.



# Home Hospice Care

- Solve with Benders decomposition.
  - **Assign aides to patients** in master problem.
    - Maximize number of patients served by a given set of aides.
  - **Schedule home visits** in subproblem.
    - Cyclic weekly schedule.
    - No visits on weekends.
  - Subproblem **decouples** into a scheduling problem for each aide and each day of the week.



# Master Problem

$\delta_j$  = 1 if patient  $j$  assigned to aide  $i$

$\delta_j$  = 1 if patient  $j$  scheduled

$$\max \sum_j \delta_j$$

$x_{ij}$  = 1 if patient  $j$  assigned to aide  $i$  on day  $k$

$$\sum_i x_{ij} = \delta_j, \quad \text{all } j$$

Required number of visits per week

$$\sum_{i,k} y_{ijk} = v_j \delta_j, \quad \text{all } j$$

$$y_{ijk} \leq x_{ij}, \quad \text{all } i, j, k$$

Spacing constraints on visit days

Benders cuts

Relaxation of subproblem

$$\delta_j, x_{ij}, y_{ijk} \in \{0, 1\}$$

# Master Problem

- For a rolling schedule:
  - Schedule **new patients**, drop **departing patients** from schedule.
    - Provide continuity for remaining patients as follows:
  - Old patients served by **same aide** on **same days**.
    - Fix  $y_{ijk} = 1$  for the relevant aides, patients, and days.

# Master Problem

- For a rolling schedule:
  - Schedule **new patients**, drop **departing patients** from schedule.
    - Provide continuity for remaining patients as follows:
  - Old patients served by **same aide** on **same days**.
    - Fix  $y_{ijk} = 1$  for the relevant aides, patients, and days.
  - Alternative: Also served **at same time**.
    - Fix time windows to enforce their current schedule.
  - Alternative: served only by **same aide**.
    - Fix  $x_{ij} = 1$  for the relevant aides, patients.



# Subproblem

Scheduling problem for aide  $i$ , day  $k$

$n$ th patient in sequence

Set of patients assigned to aide  $i$ , day  $k$

start time

$$\text{alldiff}\{\pi_n \mid n = 1, \dots, |P_{ik}|\}$$
$$[s_j, s_j + p_j] \subseteq [r_j, d_j], \quad \text{all } j \in P_{ik}$$
$$s_{\pi_n} + p_{\pi_n} + t_{\pi_n \pi_{n+1}} \leq s_{\pi_{n+1}}, \quad n = 1, \dots, |P_{ik}| - 1$$

Visit duration

Travel time

Modeled with interval variables in CP solver.

# Benders Cuts

- Generate a cut for each infeasible scheduling problem.
  - Solution of subproblem inference dual is a **proof** of infeasibility.
    - The proof may show **other** patient assignments to be infeasible.
    - Generate **nogood cut** that rules out these assignments.

# Benders Cuts

- Generate a cut for each infeasible scheduling problem.
  - Solution of subproblem inference dual is a **proof** of infeasibility.
    - The proof may show **other** patient assignments to be infeasible.
    - Generate **nogood cut** that rules out these assignments.
  - Unfortunately, we **don't have access** to infeasibility proof in CP solver.

# Benders Cuts

- So, strengthen the nogood cuts heuristically.
  - Find a smaller set of patients that create infeasibility...
    - ...by re-solving the each infeasible scheduling problem repeatedly.

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

Reduced set of patients whose assignment to aide  $i$  on day  $k$  creates infeasibility

# Benders Cuts

- Auxiliary cuts based on symmetries.
  - A cut for valid for aide  $i$ , day  $k$  is also valid for aide  $i$  on other days.
    - This gives rise to a large number of cuts.
  - The auxiliary cuts can be summed without sacrificing optimality.
    - Original cut ensures convergence to optimum.
    - This yields 2 cuts per aide:

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

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

# Subproblem Relaxation

- Include relaxation of subproblem in the master problem.
  - Necessary for good performance.
  - Use **time window relaxation** for each scheduling problem.
  - Simplest relaxation for aide  $i$  and day  $k$ :

$$\sum_{j \in J(a,b)} p_j y_{ijk} \leq b - a$$

↑  
Set of patients whose time window fits in interval  $[a, b]$ .

Can use several intervals.

# Subproblem Relaxation

- This relaxation is very weak.
  - Doesn't take into account travel times.

# Subproblem Relaxation

- This relaxation is very weak.
  - Doesn't take into account travel times.
- Improved relaxation.
  - **Basic idea:** Augment visit duration  $p_j$  with travel time to (or from) location  $j$  from **closest** patient or aide home base.



# Subproblem Relaxation

- This relaxation is very weak.
  - Doesn't take into account travel times.
- Improved relaxation.
  - **Basic idea:** Augment visit duration  $p_j$  with travel time to (or from) location  $j$  from **closest** patient or aide home base.
  - This is **weak** unless most assignments are **fixed**.
    - As in rolling schedule.

# Subproblem Relaxation

- This relaxation is very weak.
  - Doesn't take into account travel times.
- Improved relaxation.
  - **Basic idea:** Augment visit duration  $p_j$  with travel time to (or from) location  $j$  from **closest** patient or aide home base.
  - This is **weak** unless most assignments are **fixed**.
    - As in rolling schedule.
  - We partition day into 2 intervals.
    - Morning and afternoon.
    - Simplifies handling of aide time windows and home bases.
    - All patient time windows are in morning or afternoon.

# Subproblem Relaxation

Time window relaxation for aide  $i$ , day  $k$   
using intervals  $[a,b]$ ,  $[b,c]$

$$\sum_{j \in J(a,b)} p'_{ijk} y_{ijk} \leq b - a$$

$$\sum_{j \in J(b,c)} p''_{ijk} y_{ijk} \leq c - b$$

where

$[a, c]$  = time window for aide  $i$

$$p'_{ijk} = p_j + \min \left\{ t_{ij}, \min_{j' \in Q_{ik}} \{ t_{j'j} \} \right\}$$

$$p''_{ijk} = p_j + \min \left\{ \min_{j' \in Q_{ik}} \{ t_{jj'} \}, t_{ji} \right\}$$

and where  $Q_{ik} = \{\text{patients unassigned or assigned to aide } i, \text{ day } k\}$

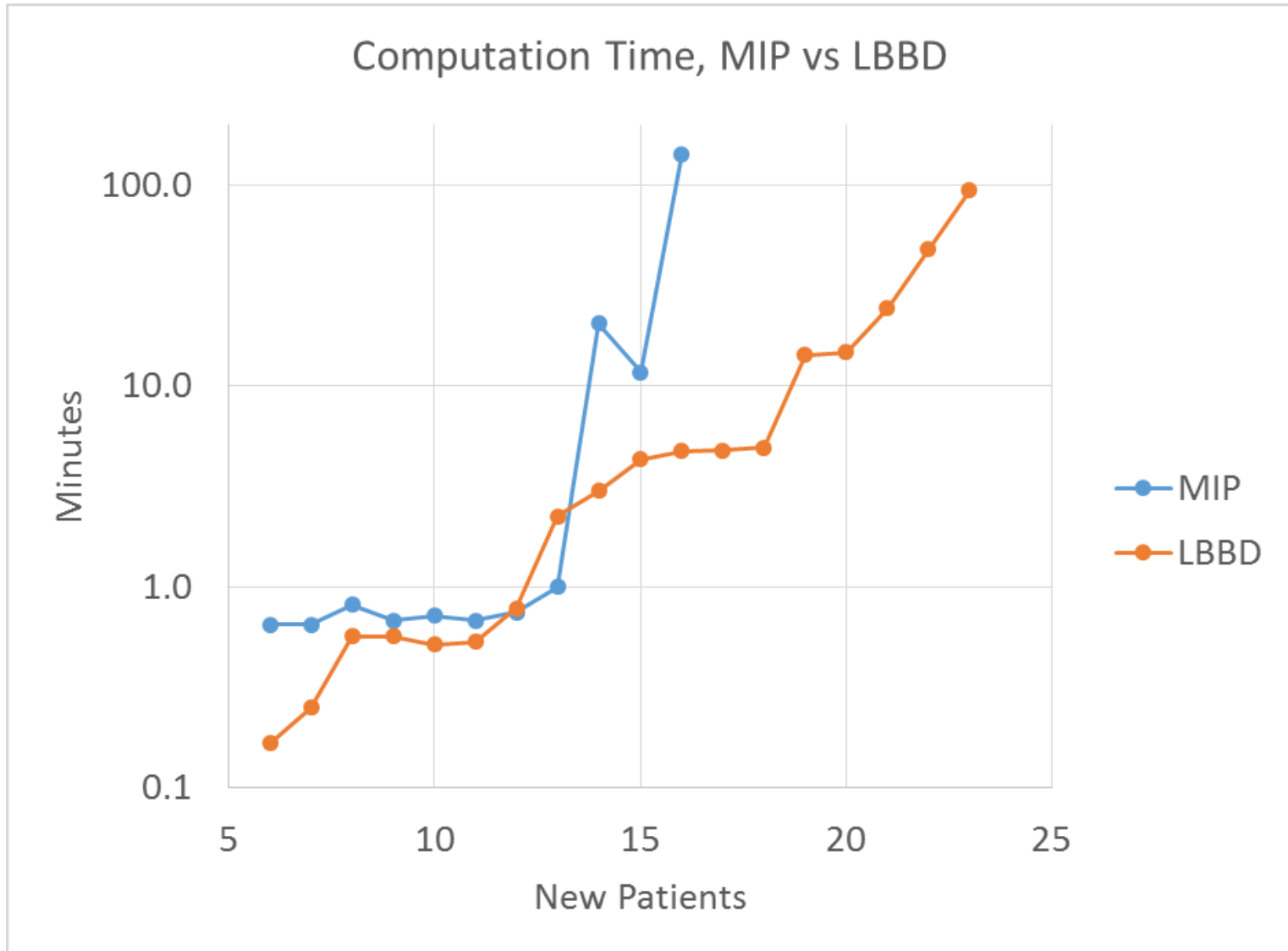
# Computational Tests

- Dataset
  - 60 home hospice patients
    - 2, 3 or 5 visits per week (not on weekends)
  - 18 health care aides with time windows
  - Actual travel distances
- Solver
  - **LBBD**: IBM ILOG Optimization Studio 12.6.2
    - CPLEX + CP Optimizer + user-supplied script
  - **MIP**: CPLEX in ILOG Studio
    - Modified multicommodity flow model of VRPTW
- Computer
  - Laptop with Intel Core i7
    - 7.75 GB RAM

# Computational Tests

- Instance generation
  - Start with (suboptimal) solution for the 60 patients
    - Fix this schedule for first  $n$  patients.
    - Schedule remaining  $60 - n$  patients
  - Use 8 of the 18 aides to cover new patients
    - As well as the old patients they already cover.
    - This puts us near the phase transition.

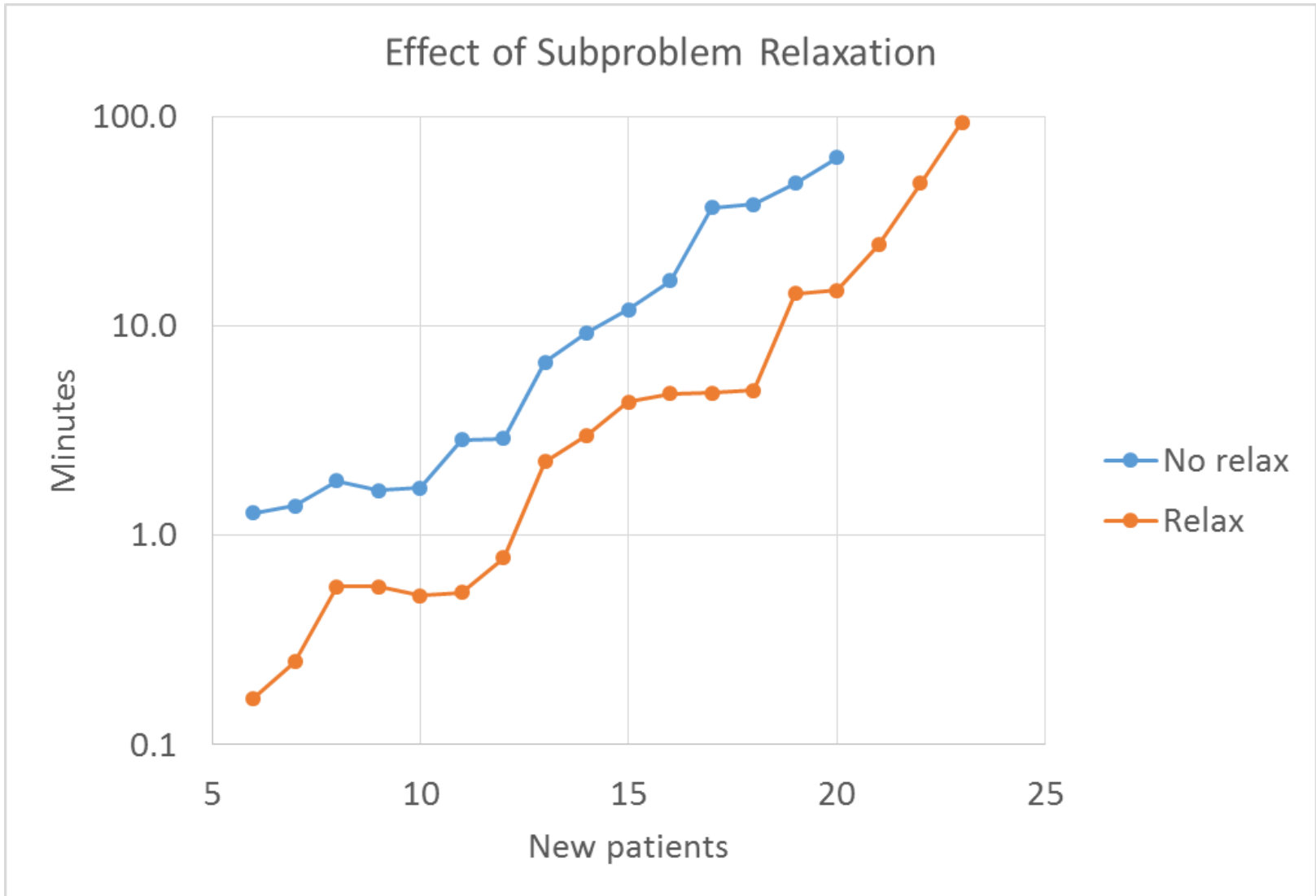
# Computational Tests



# Computational Tests

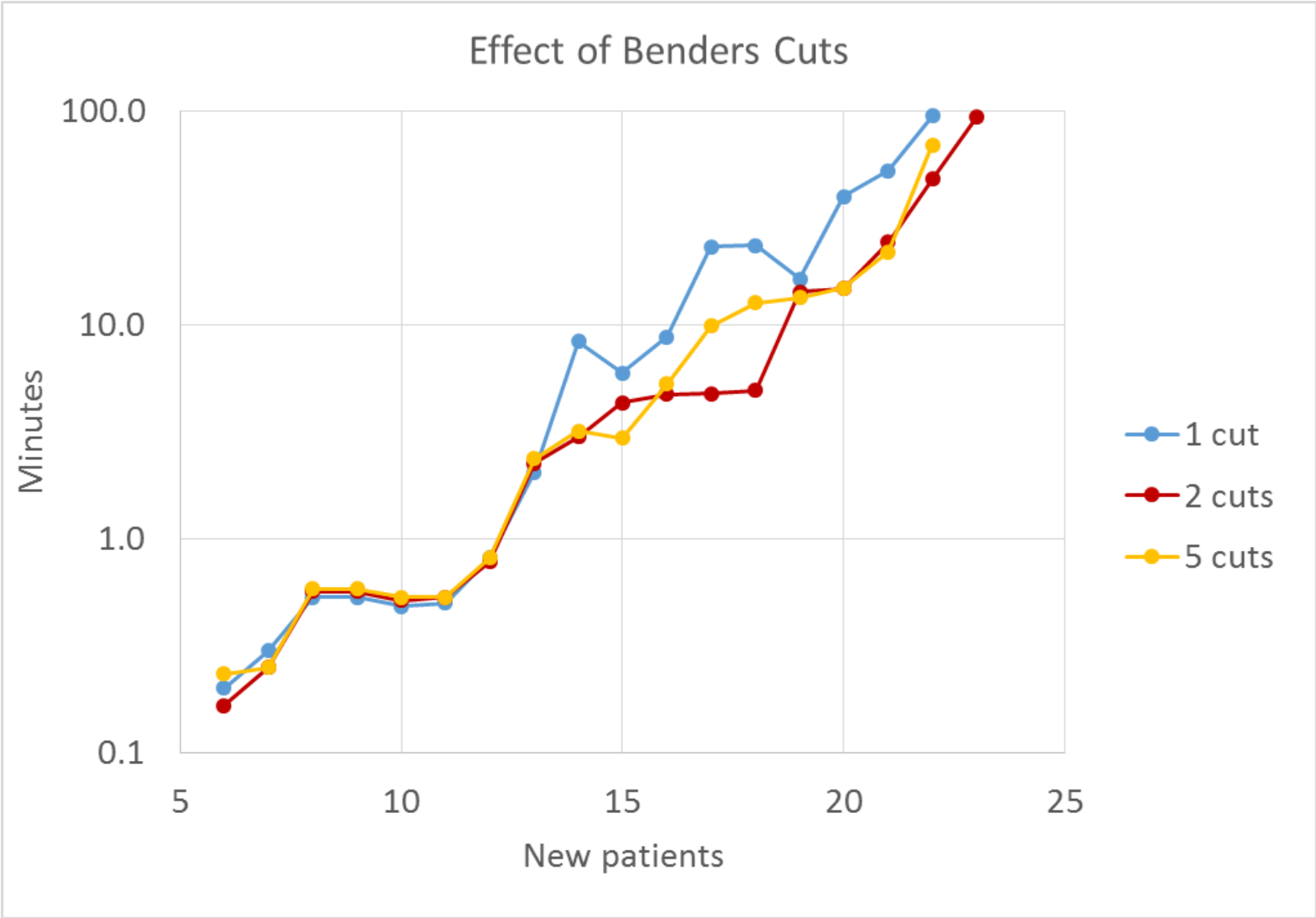
- Practical implications
  - MIP or LBBD will work for smaller instances
  - LBBD **scales up** to realistic size
    - One month advance planning in 60 patient population
    - Assuming 5-8% weekly turnover
  - Advantage of **exact** solution method
    - We know **for sure** whether existing staff will cover projected demand.

# Computational Tests





# Computational Tests



# Computational Tests

- Other relaxations
  - Multicommodity flow relaxation
    - Master problem too large, solves slowly
    - $n^2$  flow variables, where  $n$  = number of patients
    - Master must be re-solved in each iteration
    - Relaxation useless until many variables are fixed in B&B

# Computational Tests

- Other relaxations
  - Multicommodity flow relaxation
    - Master problem too large, solves slowly
    - $n^2$  flow variables, where  $n$  = number of patients
    - Master must be re-solved in each iteration
    - Relaxation useless until many variables are fixed in B&B
  - Assignment relaxation
    - Master problem still too large, solves slowly.
    - Relaxation very weak without separating TSP cuts.

# Computational Tests

- Other relaxations
  - Multicommodity flow relaxation
    - Master problem too large, solves slowly
    - $n^2$  flow variables, where  $n$  = number of patients
    - Master must be re-solved in each iteration
    - Relaxation useless until many variables are fixed in B&B
  - Assignment relaxation
    - Master problem still too large, solves slowly.
    - Relaxation very weak without separating TSP cuts.
  - Discrete time relaxation
    - Future research.
    - Unclear how to encode sequence-dependent times.

# Conclusions

- LBBD can scale up despite sequence-dependent costs...
  - ...when computing a **rolling** schedule
    - Time window relaxation is tight enough in this case

# Conclusions

- LBBD can scale up despite sequence-dependent costs...
  - ...when computing a **rolling** schedule
    - Time window relaxation is tight enough in this case
- Relaxation is key
  - Relaxation that grows quadratically is too large
    - Such as multicommodity flow and assignment relaxations
  - Relaxation must grow only linearly
    - Such as time window relaxation
    - Will try discrete time relaxation

# Conclusions

- LBBD can scale up despite sequence-dependent costs...
  - ...when computing a **rolling** schedule
    - Time window relaxation is tight enough in this case
- Relaxation is key
  - Relaxation that grows quadratically is too large
    - Such as multicommodity flow and assignment relaxations
  - Relaxation must grow only linearly
    - Such as time window relaxation
    - Will try discrete time relaxation
- Auxiliary Benders cuts can help
  - Based on subproblem symmetries
    - Good idea to aggregate auxiliary cuts

# Ongoing Research

- **Try branch and check**
  - Variant of LBBD
    - Solve master once.
    - Generate Benders cuts at some nodes of search tree
    - **Not** the same as branch and cut!
  - Allows stronger relaxation to be used in the master
    - Multicommodity flow

JH 2000; Thorsteinsson 2001



# Ongoing Research

- **Try branch and check**
  - Variant of LBBD
    - Solve master once.
    - Generate Benders cuts at some nodes of search tree
    - **Not** the same as branch and cut!
  - Allows stronger relaxation to be used in the master
    - Multicommodity flow
- **Develop practical tool for the company**
  - Additional constraints
  - Practical interface for users in several cities
  - Daily as well as long-term updates