THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

Exact methods for solving assembly line balancing problems

Joel Trouchet

A thesis submitted for the degree of Bachelor of Advanced Science (Honours) at The University of Queensland in 2025

School of Mathematics and Physics

Abstract

Assembly line balancing problems (ALBPs) involve assigning tasks to stations along an assembly line while satisfying precedence constraints. Increasing production complexity, such as multi-sided layouts and human-robot collaboration, has created variants that are challenging to solve efficiently with existing exact optimisation methods. Mixed-integer linear programming (MILP) remains the most common exact approach; however, many MILP formulations rely on Big-M constraints and exhibit symmetry, often resulting in slow solve times and difficulty proving optimality. Recent advances in constraint programming (CP) suggest it may offer a stronger alternative for highly constrained scheduling problems.

This thesis examines six ALBP variants: the simple problem (SALBP-1), two two-sided problems (TALBP-1, TALBP-2), and three human-robot collaboration problems (HRALBP-1, HRALBP-2, HRALBP-3). State-of-the-art MILP models from the literature were catalogued, analysed, and improved by strengthening time and precedence constraints and reducing symmetry. New CP formulations were then developed to exploit the built-in scheduling constraints of the chosen CP solver and reduce or eliminate Big-M constraint usage.

Computational experiments on benchmark datasets demonstrate that the improved MILP models outperform existing formulations, reducing solution time and increasing optimality rates. However, CP consistently outperforms both baseline and improved MILP models. For example, in TALBP-1, CP solved all benchmark instances optimally, while MILP variants solved only 10 and 18 of the same 35 instances.

Overall, this thesis shows that CP provides a more scalable exact optimisation framework for complex assembly line balancing problems and introduces a unified modelling approach that supports multiple ALBP variants.

Declaration by author

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

I have clearly stated the contribution of others to my thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, financial support and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of my higher degree by research candidature and does not include a substantial part of work that has been submitted to qualify for the award of any other degree or diploma in any university or other tertiary institution. I have clearly stated which parts of my thesis, if any, have been submitted to qualify for another award.

I acknowledge that an electronic copy of my thesis must be lodged with the University Library and, subject to the policy and procedures of The University of Queensland, the thesis be made available for research and study in accordance with the Copyright Act 1968 unless a period of embargo has been approved by the Dean of the Graduate School.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate I have obtained copyright permission from the copyright holder to reproduce material in this thesis and have sought permission from co-authors for any jointly authored works included in the thesis.

Contributions by others to the thesis

No contributions by others.

Financial support

No financial support was provided to fund this research.

Contents

List of Figures

List of Tables

List of abbreviations

Set	Description
N	Set of tasks to be performed
S	Set of available stations
C	Set of admissible cycle times (SALBP-2)
$D = \{1,2\}$	Set of assembly line directions (U-shaped problems)
$K = \{1,2\}$	Set of sides of the assembly line (two-sided problems)
W_s	Subset of tasks that can be assigned to station $s \in S$
W_{sk}	Subset of tasks that can be assigned to station $s \in S$ on side $k \in K$ (two-sided problems)
P_i	Subset of tasks that are immediate predecessors of task $i \in N$
F_i	Subset of tasks that are immediate followers of task $i \in N$
P_i^a	Subset of tasks that are predecessors of task $i \in N$
F_i^a	Subset of tasks that are followers of task $i \in N$
PF_i	Subset of all predecessors and followers of task $i \in N$
$P_0 = \{i \in N \mid P_i = \emptyset\}$	Subset of tasks that have no immediate predecessors
K_i	Set of sides task $i \in N$ can be performed on (two-sided problems)
N^L	Subset of tasks required to be performed on the left side of a station (two-sided problems)
N^R	Subset of tasks required to be performed on the right side of a station (two-sided problems)
N^E	Subset of tasks required to be performed on either side of a station (two-sided problems)
N_i^O	Subset of tasks whose operation side are opposite to the operation side of task $i \in N$ (two-sided problems)
R_i	Subset of tasks that are neither predecessors or followers of task $i \in N$ (two-sided problems)
PC	Set of pairs of tasks and predetermined station positional restrictions (TALBP-2)
PZ	Set of pairs of tasks with positive zoning restriction (TALBP-2 and HRALBP-3)
NZ	Set of pairs of tasks with negative zoning restriction (TALBP-2 and HRALBP-3)
SC	Set of pairs of tasks with synchronism restriction (TALBP-2)
PF_i^0	Subset of tasks that are not predecessors or followers of task $i \in N$ (human-robot problems)
H	Set of humans (human-robot problems)
R	Set of robots (human-robot problems)
NC	Set of pairs of tasks and stations they cannot be performed in (HRALBP-3)
JT	Set of pairs of tasks that have to be performed with human and robot collaboration (HRALBP-3)
TC	Set of pairs of tasks that require a robot tool change between them being performed (HRALBP-3)

Table 1: Set notation.

Data	Description
$n = \|N\|$	Number of tasks to be performed
$m = \|S\|$	Number of available stations
c	Maximum cycle time (Type-1 problems)
$\bar{c} = \max C$	Upper bound on cycle time
$\underline{c} = \min C$	Lower bound on cycle time
M	Large positive number
ε	Small positive number
t_i	Execution time of task $i \in N$
E_i	Earliest station to which task $i \in N$ can be assigned
L_i	Latest station to which task $i \in N$ can be assigned
E_{ic}	Earliest station to which task $i \in N$ can be assigned given a cycle time c (SALBP-2)
L_{ic}	Latest station to which task $i \in N$ can be assigned given a cycle time c (SALBP-2)
$n^h = \|H\|$	Maximum number of humans that can be assigned to each station (human-robot problems)
$n^r = \|R\|$	Maximum number of robots that can be assigned to each station (human-robot problems)
t_{ih}^h	Execution time of task $i \in N$ by human $h \in H$ (human-robot problems)
t_{ir}^r	Execution time of task $i \in N$ by human $r \in R$ (human-robot problems)
TC_m	Maximum tool changes allowed each robot at each station (HRALBP-3)
c_s^s	Cost of station s based on space and equipment costs per year (HRALBP-3)
c_h^h	Cost of human h based on yearly salary (HRALBP-3)
c_r^r	Cost of robot r based on purchase cost per year (HRALBP-3)
e_{ir}	Cost per year of energy consumption robot r to perform task i (HRALBP-3)

Table 2: Data notation.

Notation	Description
$O = \{1,2\}$	set of operator types (1 is humans, 2 is robots)
$L = \{1,2\}$	set of operators
n_o^l	maximum number of operators of type o that can be assigned to each station
t_{ilo}^l	time to complete task i for operator l of type o
c_{lo}^l	Cost of operator l of type o (yearly salary for humans, purchase cost per year for robots)

Table 3: Additional sets and data notation for human-robot CP formulations.

Chapter 1

Introduction

Operations research (OR) is a discipline focused on the application of mathematical modelling, optimisation, and computational methods to complex decision-making problems. One of its most prominent applications in manufacturing is the assembly line balancing problem (ALBP), the challenge of assigning tasks to a sequence of workstations in a way that meets precedence constraints while optimising key performance metrics.

The concept of the assembly line was revolutionised by the Ford Motor Company's introduction of the moving assembly line, which allowed for the systematic division of labour and significantly improved manufacturing efficiency. In modern production systems, the complexity of products and processes has only grown, making efficient assembly line balancing critical for minimising idle time, reducing costs, and maximising throughput.

The assembly line balancing problem (ALBP) is an optimisation problem with numerous variants. This thesis catalogues four prominent types: the simple assembly line balancing problem (SALBP), the U-shaped assembly line balancing problem (UALBP), the two-sided assembly line balancing problem (TALBP) and the assembly line balancing problem with human-robot collaboration (HRALBP). Each problem type can be divided into two variants: Type-1 (minimising the number of workstations for a given cycle time) and Type-2 (minimising the cycle time for a given number of workstations).

1.1 Exact and heuristic solution methods

Problems in OR are generally solved using one of two types of methods: exact or heuristic. The primary difference between these is their accuracy and efficiency. Exact methods aim to find the optimal solution to a problem; however, their performance suffers for large-scale problems. Heuristics aim to find a good solution quickly but cannot guarantee an optimal solution Zheng et al. (2024). In my research, I aim to compare two exact methods: mixed-integer linear programming (MILP) and constraint programming (CP).

MILP is an extension of Linear Programming (LP) that allows some or all decision variables to take integer or binary values, in addition to continuous ones (Zafar et al., 2018). Like LP, MILP optimises a linear objective function subject to linear equality and inequality constraints (Fachrizal et al., 2020). It is particularly useful for representing systems that involve discrete decisions, such as on/off operations, scheduling choices, or logical constraints.

MILP problems are generally solved using branch-and-bound or related decomposition algorithms that combine LP relaxations with integer search strategies (Ren and Sun, 2023). While more computationally intensive than standard LP, MILP enables more realistic and flexible modelling of practical problems.

Similar to MILP, CP is a form of mathematical programming used to solve combinatorial optimisation problems (Rossi et al., 2006). However, it is especially useful for highly constrained problems, due to its ability to utilise both logical constraints and linear constraints (Froger et al., 2016). The CP solver used in this thesis (Google OR-Tools CP-SAT) is particularly effective when applied to scheduling problems, due to its built-in scheduling constraints.

1.2 Simple assembly line balancing problem (SALBP)

The simple assembly line balancing problem (SALBP) is the most fundamental and widely studied form of the assembly line balancing problem (Boysen et al., 2022). In this formulation, tasks are arranged along a single, linear sequence of workstations. Each task has a fixed processing time, and precedence constraints must be respected, ensuring that specific tasks are completed before others can begin. The objective is to assign tasks to workstations in a manner that optimises production efficiency.

Two primary variants of SALBP are considered. SALBP-1 aims to minimise the number of workstations required to achieve a predetermined cycle time. This variant is particularly relevant when the production rate is fixed, and manufacturers wish to reduce labour and equipment costs. SALBP-2, on the other hand, seeks to minimise the cycle time for a given number of workstations, which is applicable when factory space or resource availability limits the number of stations. Despite its simplicity, the SALBP is known to be NP-hard (Álvarez Miranda and Pereira, 2019), and exact methods such as mixed-integer linear programming (MILP) have been widely applied to solve it.

1.3 U-shaped assembly line balancing problem (UALBP)

The U-shaped assembly line balancing problem (UALBP) extends the classical SALBP by arranging the line in a U-shape (Figure 1.1). This layout allows tasks to be assigned to both the forward and backward segments of the line, offering increased flexibility in task assignments and improved workstation utilisation. U-shaped lines are commonly used in lean manufacturing environments where space efficiency and operator versatility are prioritised.

As with the SALBP, the UALBP includes two key variants: UALBP-1 (minimising the number of workstations given a fixed cycle time) and UALBP-2 (minimising the cycle time given a fixed number of workstations). The additional flexibility of assigning tasks to both directions of the line increases the solution space and complexity, often requiring more advanced formulations and solution methods.

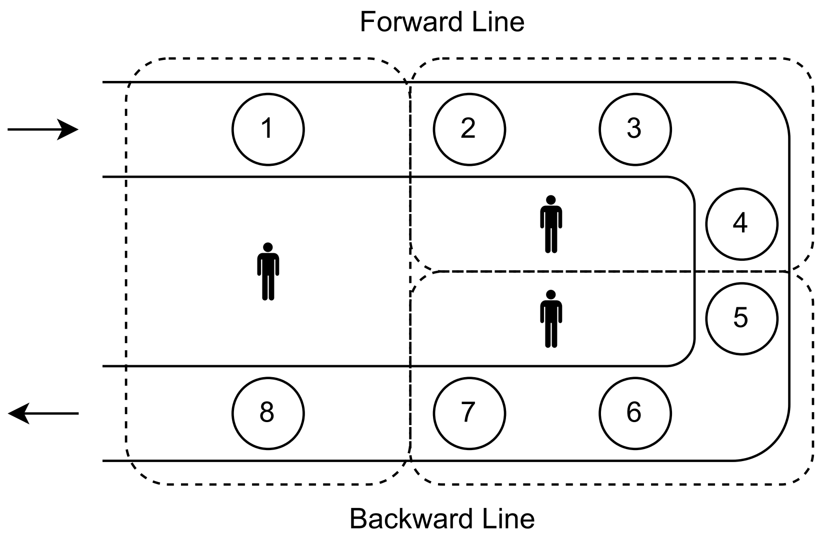

Figure 1.1: Example layout of a U-shaped assembly line. The circles indicate tasks to be performed, and the dotted rectangles indicate stations. A workpiece will first flow along the forward line, then back along the backward line.

1.4 Two-sided assembly line balancing problem (TALBP)

The two-sided assembly line balancing problem (TALBP) models assembly lines where workstations are positioned on both the left and right sides of a line, allowing specific tasks to be performed on either side (Figure 1.2). This configuration is common in large product assembly, such as automotive manufacturing, where physical constraints and assembly feasibility require tasks to be allocated to one side or the other. The problem is further complicated by the need to avoid interference between opposite-side tasks and by synchronisation constraints.

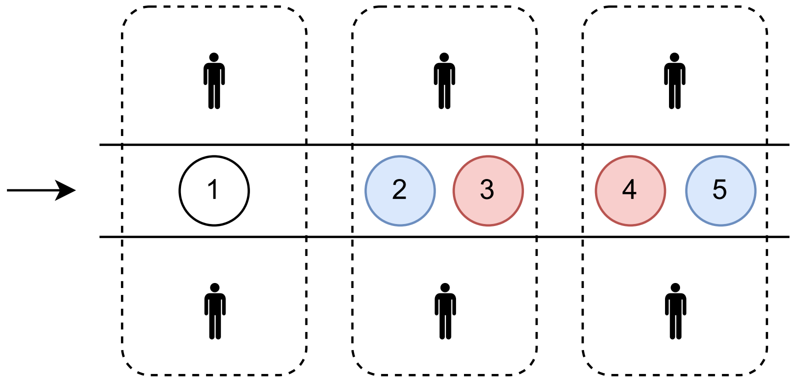

Figure 1.2: Example layout of a two-sided assembly line. Blue circles indicate tasks required to be performed on the left side of the line, red circles indicate tasks required to be performed on the right side of the line, and white circles indicate tasks that may be performed on either side. Dotted rectangles indicate mated stations.

Similar to the previous problem types, TALBP is divided into two variants: TALBP-1 and TALBP-2. TALBP-1 focuses on minimising the number of workstations for a given cycle time, while TALBP-2 aims to minimise the cycle time given a fixed number of workstations. The complexity of task assignment decisions in TALBP is significantly higher than in SALBP and UALBP, due to side constraints, incompatibility relations, and precedence restrictions.

1.5 Human-robot collaboration (HRALBP)

The assembly line balancing problem with human-robot collaboration (HRALBP) aims to further model the utility of assembly lines by considering lines where humans and robots can work together on a workpiece. Since the advent of automated manufacturing, such an assembly line has been common in large-scale manufacturing operations. The problem is complicated as it involves choosing between sets of humans and robots to perform each task, with each human and robot requiring a different amount of time to complete each task. Figure 1.3 shows a solution to a simple example of such a problem. Separating a single line into four separate lines, as shown, makes visualising results simpler; however, in reality, this would appear as multiple humans and/or multiple robots at a single station along the line.

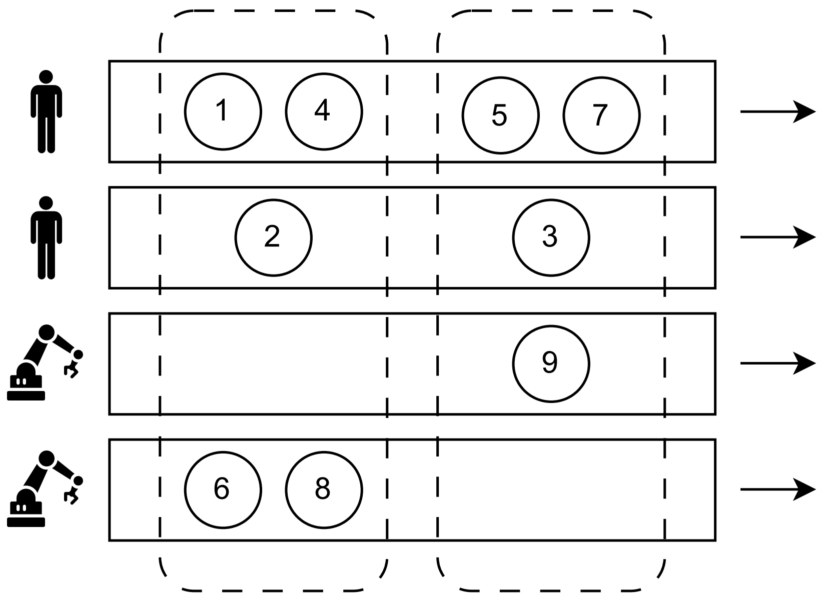

Figure 1.3: Example layout of an assembly line with human-robot collaboration. In this example, task allocation can be chosen from two humans and two robots. Circles indicate tasks, and dotted rectangles indicate stations on the line.

As with the previous problem types, HRALBP is separated into two variants, HRALBP-1 and HRALBP-2. As before, these aim to minimise the number of stations given a set cycle time and minimise the cycle time given a set number of workstations. However, there is also a third variant, HRALBP-3. The purpose of this variant is to minimise setup costs for the whole line, taking into account worker salary, robot purchasing cost, equipment cost for each station, and the cost of energy consumption for each robot. Additionally, HRALBP-3 considers complex requirements such as zoning restrictions and joint task collaboration between humans and robots.

1.6 Thesis objective and contributions

In this thesis, I aim to develop alternative exact formulations and compare them with current state-of-the-art formulations across these four problem variants and their respective types. By conducting a detailed comparative analysis, this research will contribute to understanding the potential of constraint programming as a competitive or complementary alternative to mixed-integer linear programming in addressing the complexities of modern assembly line balancing problems.

I have contributed to the field of research by providing a collated framework of state-of-the-art exact formulations for the above problems, by ensuring cohesive notation across all formulations. Furthermore, I have developed alternative MILP and CP formulations for many of them. In all cases, the new CP formulations achieved significantly shorter computation times over the previous state-of-the-art formulations, highlighting the strengths of constraint programming for these types of problems.

Chapter 2

Literature review

In this chapter, I introduce the current state-of-the-art MILP techniques for solving the problems outlined in Chapter 1. Tables 1 and 2 respectively outline the set and data notation that are used in each of the following formulations. The following sets are defined as:

$$PF_i^0 = \{j : j \in N - PF_i \wedge i \geq j\}, \qquad \forall i \in N$$
$$N_i^O = \begin{cases} N^L, & \text{if } i \in N^R \\ N^R, & \text{if } i \in N^L \\ \emptyset, & \text{if } i \in N^E \end{cases}, \qquad \forall i \in N$$
$$R_i = \left\{ j | j \in N - \left(P_i^a \cup F_i^a \cup N_i^O \right) \wedge i < j \right\}, \qquad \forall i \in N$$
$$K_i = \begin{cases} \{1\}, & \text{if } i \in N^R \\ \{2\}, & \text{if } i \in N^L \\ \{1,2\}, & \text{if } i \in N^E \end{cases}, \qquad \forall i \in N$$

2.1 SALBP-1

The aim of the simple assembly line balancing problem Type-1 (SALBP-1) is to minimise the number of stations required to complete a given set of tasks. Pastor et al. (2011) outlined seven models to optimise SALBP-1, denoted M1 through M7. For this thesis, all but M2 are included, as it utilises an extra "dummy" task, which complicates the formulation's interpretation while providing no substantial computational benefits, as outlined in Pastor et al. (2011).

To track which tasks are completed at each workstation, and which workstations are used, two binary decision variables were defined as shown in Table 2.1.

Notation	Description	Defined for
$x_{is} \in \{0,1\}$	1 if task i is completed at station s, 0 otherwise	$i \in N,\ s \in [E_i, L_i]$
$y_s \in \{0,1\}$	1 if station s is used, 0 otherwise	$s \in S$

Table 2.1: Variable definitions for SALBP-1.

A basic SALBP-1 model is formulated as:

$$\min \quad \sum_{s\in S} y_s \tag{2.1}$$

s.t.

$$\sum_{s\in[E_i,L_i]} x_{is} = 1, \qquad \forall i \in N \tag{2.2}$$

$$\sum_{i\in W_s} t_i \cdot x_{is} \leq c, \qquad \forall s \in S \tag{2.3}$$

$$\sum_{s\in[E_i,L_i]} s \cdot x_{is} \leq \sum_{s\in EL_j} s \cdot x_{js}, \qquad \forall i \in N,\ j \in F_i \tag{2.4}$$

$$\sum_{i\in W_s} x_{is} \leq |W_s| \cdot y_s, \qquad \forall s \in S \tag{2.5}$$

Equation (2.1) contains the objective function in which the number of stations used is minimised. Constraint (2.2) restricts the task assignment such that a task can only be assigned to a single station, and can only be assigned to a station within its allowed station subset. Constraint (2.3) ensures that all task assignments satisfy the maximum cycle time. Constraint (2.4) constrains the precedence relations in the formulation, ensuring a task i will be completed before or in the same station as its immediate follower task j. Constraint (2.5) connects the two binary decision variables by restricting the number of tasks performed on a single station to the size of the subset of tasks that may be performed at that station.

The use of the objective function in Equation (2.1) creates large symmetry groups within the formulation, making it difficult for the problem to be solved. Hence, Pastor et al. (2011) proposed the objective function in Equation (2.6) to break the symmetry in the problem (Margot, 2010).

$$\min \quad \sum_{s\in S} s \cdot y_s \tag{2.6}$$

In its current state, the formulation will allow stations along the assembly line to be activated without previous stations first being activated, resulting in large gaps between used stations. Furthermore, this also creates symmetry within the station assignment constraints. To break this symmetry, Pastor et al. (2011) proposed additional Constraints (2.7) and (2.8). Constraint (2.7) ensures that stations $s+1, s+2, ..., m$ cannot be activated unless station s is activated, and Constraint (2.8) ensures that station $s+1$ cannot be activated unless station s is activated.

$$y_s \geq y_u, \qquad \forall s \in S \setminus \{m\},\ u \in [s+1, m] \tag{2.7}$$

$$y_s \geq y_{s+1}, \qquad \forall s \in S \setminus \{m\} \tag{2.8}$$

With these constraints, Table 2.2 summarises the equations that make up each model.

Base Model	(2.2)-(2.5)	
Model	Objective	Constraints
M1	(2.1)	
M3 (symmetries broken by objective function)	(2.6)	
M4 (symmetries broken by constraints)	(2.1)	(2.7)
M5 (symmetries broken by constraints)	(2.1)	(2.8)
M6 (symmetries broken by objective function and constraints)	(2.6)	(2.7)
M7 (symmetries broken by objective function and constraints)	(2.6)	(2.8)

Table 2.2: SALBP-1 models. Each objective and constraint equation listed is in addition to the base model equations.

2.2 SALBP-2

The aim of the simple assembly line balancing problem Type-2 (SALBP-2) is to minimise the maximum time a workpiece spends at an individual station (cycle time), given a set number of workstations. Ritt and Costa (2018) presented 16 formulations to solve this type of problem, 12 of which were tested on a benchmark data set.

To set appropriate bounds for the stations in which tasks can be assigned, the authors presented the following definitions for the earliest and latest station tasks can be assigned to, respectively, E_{ic} and L_{ic}.

$$E_{ic} = \left\lceil \sum_{j \in N | j \leq i} \frac{t_j}{c} \right\rceil \qquad \forall i \in N,\ c \in C$$

$$L_{ic} = m - 1 - \left\lceil \sum_{j \in N | i \leq j} \frac{t_j}{c} \right\rceil \qquad \forall i \in N,\ c \in C$$

Furthermore, the authors also determined the appropriate bounds for cycle time to be as follows.

$$\underline{c} = \max \left\{ \max_{i \in N} t_i, \left\lceil \sum_{i \in N} \frac{t_i}{m} \right\rceil \right\}$$

$$\bar{c} = 2\underline{c}$$

$$C = [\underline{c}, \bar{c}]$$

2.2.1 Formulation

A basic formulation of SALBP-2 seeks to minimise cycle time while obeying occurrence and precedence constraints. This basic formulation utilises the binary decision variable x_{is} and the continuous variable c, where c represents the minimised cycle time of the problem solution.

$$x_{is} \in \{0,1\}, 1 \text{ if task } i \text{ is assigned to workstation } s, 0 \text{ otherwise}, \qquad \forall i \in N,\ s \in S \tag{2.9}$$

$$c \in \mathbb{R}, \quad \text{cycle time} \tag{2.10}$$

The SALBP-2 formulation is as follows:

$$\min \quad c \tag{2.11}$$

s.t.

$$\sum_{i \in N} t_i \cdot x_{is} \leq c, \qquad \forall s \in S \tag{2.12}$$

$$\sum_{s \in S} x_{is} = 1, \qquad \forall i \in N \tag{2.13}$$

$$x_{js} \leq \sum_{u \in S | u \leq s} x_{iu}, \qquad \forall i \in N,\ j \in F_i,\ s \in S \tag{2.14}$$

The objective function in Equation (2.11) represents a minimisation of the cycle time in the formulation. Constraint (2.12) connects the task assignment variable to the cycle time, and Constraint (2.13) is a single assignment constraint, ensuring that a task is assigned to a single station. Constraint (2.14) ensures precedence relations are satisfied. An immediate issue with this formulation is the lack of imposed station limits. In its current state, task assignments can occur outside of their station bounds. To remedy this issue, the following definition for the binary decision variable x_{is} is used instead of the one defined in Equation (2.9).

$$x_{is} \in \{0,1\}, 1 \text{ if task } i \text{ is assigned to workstation } s, 0 \text{ otherwise}, \qquad \forall i \in N,\ s \in [E_{i\bar{c}}, L_{i\bar{c}}] \tag{2.15}$$

Additionally, to strengthen the station bounds, Ritt and Costa (2018) proposed a second binary decision variable to model the cycle time explicitly as shown in Equation (2.16).

$$r_t \in \{0,1\}, 1 \text{ if cycle time } t \text{ is used, } 0 \text{ otherwise}, \qquad \forall t \in C \tag{2.16}$$

The problem can be reformulated with the following additional constraints:

$$c = \sum_{t \in C} t \cdot r_t \tag{2.17}$$

$$\sum_{t \in C} r_t = 1 \tag{2.18}$$

$$x_{is} \leq 1 - \sum_{t \in C | s < E_{it}} r_t, \qquad \forall i \in N,\ s \in [E_{i\bar{c}}, E_{i\underline{c}}) \tag{2.19}$$

$$x_{is} \leq 1 - \sum_{t \in C | L_{it} < s} r_t, \qquad \forall i \in N,\ s \in (L_{i\underline{c}}, L_{i\bar{c}}] \tag{2.20}$$

Constraint (2.17) assigns the explicit cycle time to the cycle time variable, Constraint (2.18) ensures that only one explicit cycle time can be chosen, and Constraints (2.19) and (2.20) apply the stations limits to task assignments. In these constraints, when a task cannot be assigned to a station earlier than e, this constraint also holds for all stations preceding e. The same fact holds for all stations following l, when a task cannot be assigned to a station later than l. Hence, these constraints can be strengthened as follows.

$$\sum_{u \in S | u \leq s} x_{iu} \leq 1 - \sum_{t \in C | s < E_{it}} r_t, \qquad \forall i \in N,\ s \in [E_{i\bar{c}}, E_{i\underline{c}}) \tag{2.21}$$

$$\sum_{u \in S | u \geq s} x_{iu} \leq 1 - \sum_{t \in C | L_{it} < s} r_t, \qquad \forall i \in N,\ s \in (L_{i\underline{c}}, L_{i\bar{c}}] \tag{2.22}$$

In its current form, precedence Constraint (2.14) allows for symmetry within the formulation. So, the authors proposed three alternatives, as shown in Constraints (2.23) to (2.25), to address this issue.

$$\sum_{s \in S} s \cdot x_{is} \leq \sum_{s \in S} s \cdot x_{js}, \qquad \forall i \in N,\ j \in F_i \tag{2.23}$$

$$\sum_{s \in S} (m - s + 1)(x_{is} - x_{js}) \geq 0, \qquad \forall i \in N,\ j \in F_i \tag{2.24}$$

$$\sum_{u \in S | u \leq s} x_{iu} \geq \sum_{u \in S | u \leq s} x_{ju}, \qquad \forall i \in N,\ j \in F_i,\ s \in S \tag{2.25}$$

Constraint (2.23) breaks symmetry by imposing an ordering via weighted sums in the set of stations S (Ghoniem and Sherali, 2011). This creates an ordering within the constraint, removing some solution permutations from the formulation. Constraint (2.24) breaks symmetry by imposing a lexicographic ordering on each task $i \in N$ and its immediate followers $j \in F_i$. This assigns larger coefficients to $x_{is} - x_{js}$ for smaller values of s (i.e. as s increases, the coefficient decreases), placing a stronger weight on lower values of s. If the first s where $x_{is} \neq x_{js}$ results in a difference of $x_{is} - x_{js} = 1$, the stronger weighting ensures the ordering $i \leq j$. However, if the first difference is $x_{is} - x_{js} = -1$, the larger weight makes it harder for terms of larger s values to compensate. This results in a heavy aversion to symmetric swapping within the formulation (Margot, 2010).

Constraint (2.25) breaks symmetry by imposing cumulative ordering on each task $i \in N$ and its immediate followers $j \in F_i$. This means for each pair of tasks (i, j), the sum of task assignments are compared for $u \in S | u \leq s$, ensuring that for every $s \in S$, the sequence of x_{is} cannot "fall behind" the sequence of x_{js}. This causes a dominance condition at every $s \in S$, increasing the "cost" of swapping to different solution permutations (Chu and Stuckey, 2015).

Table 2.3 summarises the equations that make up each model.

Base Model	(2.9)-(2.13)
Precedence constraints	
Model	
BW	(2.14)
PA (symmetries broken by weighted sums)	(2.23)
TS (symmetries broken by lexicographic ordering)	(2.24)
NF (symmetries broken by cumulative ordering)	(2.25)
Station limits	
Sub-model	
1 (no station limits)	
2 (station limits in decision variable)	(2.15)
3 (station limits in explicitly modelled time)	(2.15)-(2.20)
4 (station limits with strengthened constraints)	(2.15)-(2.18), (2.21), (2.22)

Table 2.3: SALBP-2 models. Each of the four models (BW, PA, TS, NF) are to be modelled with each of the four sub-models (1, 2, 3, 4).

2.3 UALBP-1

The aim of the U-shaped assembly line balancing problem Type-1 (UALBP-1) is to minimise the number of stations required for completing a set of tasks, given a set cycle time, on a U-shaped line. U-shaped assembly lines are designed so that all tasks first pass through all stations in the forward direction and then pass through them again in the backward direction. This complicates the model as the precedence constraints must account for tasks whose predecessors have not only been completed on a preceding station, but also earlier stations in the preceding direction on the line. Ritt and Costa (2018) present three formulations to solve this problem, only one of which was tested on benchmark data.

Station limits are redefined in Equations (2.26) and (2.27). E and L are now only defined for each task $i \in N$ as the cycle time c is a known piece of data for Type-1 formulations.

$$E_i = \left\lceil \sum_{j \in N | j \leq i} \frac{t_j}{c} \right\rceil, \qquad \forall i \in N \tag{2.26}$$

$$L_i = \left\lceil \sum_{j \in N | i \leq j} \frac{t_j}{c} \right\rceil, \qquad \forall i \in N \tag{2.27}$$

2.3.1 Formulation

The UALBP-1 formulation makes use of three binary decision variables, as shown in Table 2.4.

Notation	Description	Defined for
$x_{is} \in \{0,1\}$	1 if task i is assigned to station s in the forward pass, 0 otherwise	$i \in N,\ s \in S$
$w_{is} \in \{0,1\}$	1 if task i is assigned to station s in the backward pass, 0 otherwise	$i \in N,\ s \in S$
$y_s \in \{0,1\}$	1 if station s is used, 0 otherwise	$s \in S$

Table 2.4: Variable definitions for UALBP-1.

The two variables x_{is} and w_{is} may be combined into a single variable x_{isd} where d is the direction index of task assignment. This method is used for UALBP-2 in Section 2.4.1. The UALBP-1 formulation is as follows:

$$\min \quad \sum_{s \in S} y_s$$

s.t.

$$\sum_{s \in S} x_{is} + w_{is} = 1, \qquad \forall i \in N \tag{2.28}$$

$$\sum_{i \in N} t_i \left(x_{is} + w_{is}\right) \leq c \cdot y_s, \qquad \forall s \in S \tag{2.29}$$

$$\sum_{s \in S} (m - s + 1)\left(x_{is} - x_{js}\right) \geq 0, \qquad \forall i \in N,\ j \in F_i \tag{2.30}$$

$$\sum_{s \in S} (m - s + 1)\left(w_{is} - w_{js}\right) \geq 0, \qquad \forall i \in N,\ j \in F_i \tag{2.31}$$

Constraint (2.28) is a single assignment constraint, ensuring a task can only be assigned to one station, in a single direction. Constraint (2.29) connects the station usage variable y_s to the task assignment variables x_{is} and w_{is} while also limiting the time spent in each station to be less than the cycle time c. Constraints (2.30) and (2.31) handle the forward and backward precedence relations, respectively. While these constraints do break symmetry to some degree, they can be strengthened further. Similar to in Constraint (2.25), to break symmetry further, the authors presented the following constraints in place of the current precedence constraints.

$$\sum_{u \in S | u \leq s} x_{iu} \geq \sum_{u \in S | u \leq s} x_{ju}, \qquad \forall i \in N,\ j \in F_i,\ s \in S \tag{2.32}$$

$$\sum_{u \in S | u \leq s} w_{iu} \geq \sum_{u \in S | u \leq s} w_{ju}, \qquad \forall i \in N,\ j \in F_i,\ s \in S \tag{2.33}$$

Currently, no station limits are imposed, so the binary decision variables x and w are redefined below to impose the station limits.

$$x_{is} \in \{0,1\}, \qquad \forall i \in N,\ s \in S | E_i \leq s$$
$$w_{is} \in \{0,1\}, \qquad \forall i \in N,\ s \in S | L_i \leq s$$

2.4 UALBP-2

The aim of the U-shaped assembly line balancing problem Type-2 (UALBP-2) is to minimise the cycle time required for a set of tasks to be performed, given a set number of stations. As with UALBP-1, on a U-shaped line, a workpiece will travel along a set of workstations before returning along those workstations in the opposite direction. Li et al. (2017) proposed a single formulation to solve this problem type.

2.4.1 Formulation

The UALBP-2 formulation makes use of the continuous variable c to denote the cycle time and the binary decision variable x_{isd} as defined below:

Notation	Description	Defined for
$x_{isd} \in \{0,1\}$	1 if task i is assigned to station s in direction d, 0 otherwise	$i \in N,\ s \in S,\ d \in D$
$c \in \mathbb{R}$	cycle time	

Table 2.5: Variable definitions for UALBP-2.

The UALBP-2 formulation is as follows:

$$\begin{aligned} \min \quad & c \\ \text{s.t.} \quad & \end{aligned}$$

$$\sum_{i \in N} \sum_{d \in D} t_i \cdot x_{isd} \leq c, \qquad \forall s \in S \tag{2.34}$$

$$\sum_{d \in D} \sum_{s \in S} x_{isd} = 1, \qquad \forall i \in N \tag{2.35}$$

$$\sum_{s \in S} x_{js1} \leq \sum_{s \in S} x_{is1}, \qquad \forall i \in N,\ j \in F_i \tag{2.36}$$

$$\sum_{s \in S} s \cdot x_{is1} \leq \sum_{s \in S} s \cdot x_{js1} + M \sum_{s \in S} s \cdot x_{js2}, \qquad \forall i \in N,\ j \in F_i \tag{2.37}$$

$$\sum_{s \in S} x_{is2} \leq \sum_{s \in S} x_{js2}, \qquad \forall i \in N,\ j \in F_i \tag{2.38}$$

$$\sum_{s \in S} s \cdot x_{js2} \leq \sum_{s \in S} s \cdot x_{is2} + M \sum_{s \in S} s \cdot x_{is1}, \qquad \forall i \in N,\ j \in F_i \tag{2.39}$$

Constraint (2.34) limits the time spent at each station to be less than the cycle time, effectively connecting the cycle time to the station times. Constraint (2.35) is a single assignment constraint, ensuring that each task is only performed at a single workstation, and only in one direction. Constraints (2.36) and (2.37) ensure that if a task is performed on the forward section of the line, then all its predecessors must also be performed on the forward section of the line. Similarly, Constraints (2.38) and (2.39) ensure that if a task is performed on the backward section of the line, then all of its followers must also be performed on the backward section of the line.

2.5 TALBP-1

The aim of the two-sided assembly line balancing problem Type-1 (TALBP-1) is to minimise the number of stations used, given that tasks may be performed on either side of the assembly line. To further expand on this basic model, we define a mated-station as a pair of directly facing stations. Özcan (2010) identified the need to minimise the number of mated-stations as a primary objective, and minimise the number of stations as a secondary objective. Figure 2.1 shows two solutions to the same TALBP-1 instance, with no blended objective and with a blended objective. Both solutions result in the same number of mated-stations activated; however, the blended solution activates fewer stations. This is advantageous from a planning perspective, as it minimises the number of workers needed on the assembly line, decreasing the overall labour cost of the solution.

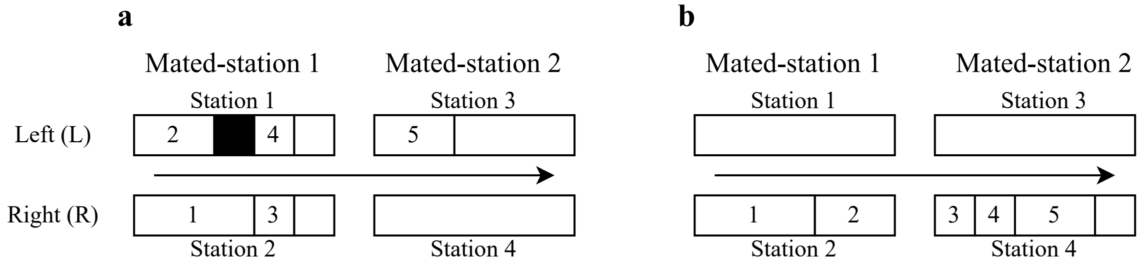

Figure 2.1: Two solutions of the same TALBP-1 Problem with maximum cycle time 5. **a** represents the simple version of the problem in which only the number of stations is minimised, and **b** represents the formulation with the blended objective. Black areas represent when the station is activated but idle.

The small positive number, ε, is used as the weighting factor for the secondary objective described above. To not cause a large perturbation in the overall objective value, Özcan (2010) defines ε within the following range:

$$0 < \varepsilon \leq \frac{1}{2n+1} \tag{2.40}$$

2.5.1 Formulation

Define the following binary and continuous variables.

Variable	Description	Defined for
$x_{isk} \in \{0,1\}$	1 if task i is assigned to workstation s on side k, 0 otherwise	$i \in N,\ s \in S,\ k \in K_i$
$u_s \in \{0,1\}$	1 if station s is used on both sides of the line, 0 otherwise	$s \in S$
$v_s \in \{0,1\}$	1 if station s is used on only one side of the line, 0 otherwise	$s \in S$
$y_{sk} \in \{0,1\}$	1 if mated-station s is used on side k	$s \in S,\ k \in K$
$z_{ij} \in \{0,1\}$	1 if task i is assigned earlier than task j in the same station	$i \in N,\ j \in R_i$
$t_i^f \in \mathbb{R}$	finish time of task i	$i \in N$

Table 2.6: Variable definitions for TALBP-1.

The TALBP-1 is formulated as follows:

$$\min \quad \sum_{s \in S} (u_s + v_s) + \varepsilon \sum_{s \in S} \sum_{k \in K} y_{sk} \tag{2.41}$$

s.t.

$$\sum_{s \in S} \sum_{k \in K_i} x_{isk} = 1, \qquad \forall i \in N \tag{2.42}$$

$$\sum_{s \in S} \sum_{k \in K_j} s \cdot x_{jsk} \leq \sum_{s \in S} \sum_{k \in K_i} s \cdot x_{isk}, \qquad \forall i \in N - P_0,\ j \in P_i \tag{2.43}$$

$$t_i^f \leq c, \qquad \forall i \in N \tag{2.44}$$

$$t_i^f - t_j^f + M\left(1 - \sum_{k \in K_j} x_{jsk}\right) + M\left(1 - \sum_{k \in K_i} x_{isk}\right) \geq t_i, \qquad \forall i \in N - P_0,\ j \in P_i,\ s \in S \tag{2.45}$$

$$t_j^f - t_i^f + M(1 - x_{isk}) + M(1 - x_{jsk}) + M(1 - z_{ij}) \geq t_j, \qquad \forall i \in N,\ j \in R_i,\ s \in S,\ k \in K_i \cap K_j \tag{2.46}$$

$$t_i^f - t_j^f + M(1 - x_{isk}) + M(1 - x_{jsk}) + M \cdot z_{ij} \geq t_i, \qquad \forall i \in N,\ j \in R_i,\ s \in S,\ k \in K_i \cap K_j \tag{2.47}$$

$$\sum_{i \in N} x_{isk} \leq |W_{sk}| \cdot y_{sk}, \qquad \forall s \in S,\ k \in K_i \tag{2.48}$$

$$\sum_{k \in K} y_{sk} = 2u_s + v_s, \qquad \forall s \in S \tag{2.49}$$

$$t_i^f \geq t_i, \qquad \forall i \in N \tag{2.50}$$

The objective function in Equation (2.41) is a blended objective, with the primary objective minimising the number of mated-stations and the secondary objective minimising the number of individual stations. Constraint (2.42) ensures that each task is only assigned to a single side of a single mated-station. Constraint (2.43) ensures that all tasks follow the precedence relations. Constraint (2.44) ensures that each task does not finish later than the given cycle time.

Constraints (2.45) - (2.47) control the finishing times of tasks. For each pair of tasks i, j, if task j is an immediate predecessor of task i, and they are assigned to the same mated-station s, then Constraint (2.45) becomes active. If two tasks do not have any precedence relations, and they are assigned to the same station, then Constraints (2.46) or (2.47) become active. If task i is assigned earlier than task j to the same station (s,k) then Constraint (2.46) becomes $t_j^f - t_i^f \geq t_j$. Otherwise, Constraint (2.47) becomes $t_i^f - t_j^f \geq t_i$.

Constraint (2.48) is the station constraint, where if a task is assigned to (s,k), then $y_{sk} = 1$, otherwise $y_{sk} = 0$. Constraint (2.49) ensures that if the left $(s,1)$ and right $(s,2)$ stations of mated-station s are used together, then $u_s = 1$ and $v_s = 0$, if only one is used, then $u_s = 0$ and $v_s = 1$, and if neither is used, then $u_s = v_s = 0$. Hence, the total number of mated-stations used in the solution is equal to the sum of u_s and v_s for all $s \in S$.

2.6 TALBP-2

The aim of the two-sided assembly line balancing problem Type-2 (TALBP-2) is to minimise cycle time on a line in which tasks may be performed on either side, given a set number of mated-stations. Li et al. (2016) proposed an extended formulation of this problem type, in which their model is designed to also account for zoning, synchronism and predetermined positional restrictions.

Zoning restrictions are divided into two types: positive and negative. Positive zoning restrictions determine the tasks that must be performed in the same station. In contrast, negative zoning restrictions determine tasks that cannot be performed in the same mated-station. This is an important inclusion, as some assembly factories may have a limited number of tools to perform specific tasks, meaning tasks that require the same tools must be performed together (positive zoning). Furthermore, there may be a capacity on the number of different tools that can be assigned to a single station, meaning tasks that require different tools cannot necessarily be performed at the same station (negative zoning).

Synchronism restrictions restrict a pair of tasks to be performed simultaneously on both sides of the same-mated station. Once again, this is an important inclusion due to the need for tasks to be performed simultaneously, such as the installation of a car axle that requires attachment to both sides of the car. Predetermined positional restrictions indicate whether a task must be allocated to a predetermined station or mated-station. This is again important as tools required to complete a specific task may only be able to be assigned to a specific station due to size constraints, for example.

2.6.1 Formulation

Since minimisation of station use is not necessary for Type-2 problems, define only the following binary and continuous variables for TALBP-2.

Variable	Description	Defined for
$x_{isk} \in \{0,1\}$	1 if task i is assigned to workstation s on side k, 0 otherwise	$i \in N,\ s \in S,\ k \in K_i$
$z_{ij} \in \{0,1\}$	1 if task i is assigned earlier than task j in the same station	$i \in N,\ j \in R_i$
$t_i^f \in \mathbb{R}$	finish time of task i	$i \in N$
$c \in \mathbb{R}$	cycle time	

Table 2.7: Variable definitions for TALBP-2.

The TALBP-2 formulation is as follows:

min c

s.t.

$$\sum_{s\in S}\sum_{k\in K_i} x_{isk} = 1, \qquad \forall i \in N \tag{2.51}$$

$$\sum_{s\in S}\sum_{k\in K_j} (s \cdot x_{jsk}) \leq \sum_{s\in S}\sum_{k\in K_i} (s \cdot x_{isk}), \qquad \forall i \in N - P_0,\ j \in P_i \tag{2.52}$$

$$t_i^f - t_j^f + M\left(1 - \sum_{k\in K_j} x_{jsk}\right) + M\left(1 - \sum_{k\in K_i} x_{isk}\right) \geq t_i, \qquad \forall i \in N - P_0,\ j \in P_i,\ s \in S \tag{2.53}$$

$$t_j^f - t_i^f + M(1 - x_{isk}) + M(1 - x_{jsk}) + M(1 - z_{ij}) \geq t_j, \qquad \forall i \in N,\ j \in R_i,\ s \in S,\ k \in K_i \cap K_j \tag{2.54}$$

$$t_i^f - t_j^f + M(1 - x_{isk}) + M(1 - x_{jsk}) + M \cdot z_{ij} \geq t_i, \qquad \forall i \in N,\ j \in R_i,\ s \in S,\ k \in K_i \cap K_j \tag{2.55}$$

$$t_i^f \leq c, \qquad \forall i \in N \tag{2.56}$$

$$x_{isk} = 1, \qquad \forall (i,(s,k)) \in PC | k \in K_i \tag{2.57}$$

$$x_{isk} - x_{jsk} = 0, \qquad \forall (i,j) \in PZ,\ k \in K_i \cap K_j,\ s \in S \tag{2.58}$$

$$\sum_{k\in K_i} x_{isk} + \sum_{k\in K_j} x_{jsk} \leq 1, \qquad \forall (i,j) \in NZ,\ s \in S \tag{2.59}$$

$$x_{isf} - x_{jsk} = 0, \qquad \forall (i,j) \in SC,\ s \in S,\ k \in K_j,\ f \in K_i | k \neq f \tag{2.60}$$

$$t_i^f - t_i = t_j^f - t_j, \qquad \forall (i,j) \in SC \tag{2.61}$$

Constraint (2.51) is the single assignment constraint, which limits each task to be performed on one side of a single station. Constraint (2.52) handles the precedence relations between tasks. Constraints (2.53)-(2.55) control the scheduling of each task. Constraint (2.56) ensures the cycle time is connected to task finishing times. Constraint (2.57) ensures the predetermined positional task restrictions are satisfied. Constraint (2.58) ensures that positive zoning task pairs are assigned to the same side of the same station, and Constraint (2.59) ensures that negative zoning task pairs are not assigned to the same mated-station. Constraint (2.60) ensures that pairs of tasks with synchronism restrictions are performed on opposite sides of the same mated-station, and Constraint (2.61) ensures that these are started at the same time.

2.7 HRALBP-1

The following formulations for HRALBP-1, HRALBP-2, and HRALBP-3 are all presented by Nourmohammadi et al. (2024). The aim of the assembly line balancing problem with human-robot collaboration Type-1 (HRALBP-1) is to minimise the number of stations, given that tasks may be performed by humans or robots.

For HRALBP, it is necessary to track when, where and by which operator (human or robot) each task is completed. Hence, define the binary variables x_{ish} and w_{isr} to decide which human and robot, respectively, is used to complete each task and at which workstation. Additionally, define the binary variables u_{ij} and v_{ij} to track all tasks completed by each human and robot, respectively, as well as the order in which they are completed. Furthermore, the station usage of each operator is tracked using the binary variable z_{sh}^h for humans and z_{sr}^r for robots. The continuous variable t_i determines the duration of each task once it has been assigned to a specific operator. These variables, as well as other required variables, are defined as follows:

Variable	Description	Defined for
$x_{ish} \in \{0,1\}$	1 if task i is assigned to human h at station s, 0 otherwise	$i \in N,\ s \in S,\ h \in H$
$w_{isr} \in \{0,1\}$	1 if task i is assigned to robot r at station s, 0 otherwise	$i \in N,\ s \in S,\ r \in R$
$u_{ij} \in \{0,1\}$	1 if task i completed before j by the same human, 0 otherwise	$i,j \in N$
$v_{ij} \in \{0,1\}$	1 if task i completed before j by the same robot, 0 otherwise	$i,j \in N$
$z_{sh}^h \in \{0,1\}$	1 if human h is used in station s, 0 otherwise	$s \in S,\ h \in H$
$z_{sr}^r \in \{0,1\}$	1 if robot r is used in station s, 0 otherwise	$s \in S,\ r \in R$
$y_s \in \{0,1\}$	1 if station s is used, 0 otherwise	$s \in S$
$t_i \in \mathbb{R}$	duration of task i with its assigned operator	$i \in N$
$t_i^s \in \mathbb{R}$	start time of task i	$i \in N$

Table 2.8: Variable definitions for HRALBP-1.

HRALBP-1 is formulated as follows:

$$\min \quad \sum_{s\in S} y_s$$

s.t.

$$\sum_{s\in S}\sum_{h\in H} x_{ish} + \sum_{s\in S}\sum_{r\in R} w_{isr} = 1, \qquad \forall i\in N \tag{2.62}$$

$$\sum_{s\in S}\sum_{h\in H} s(x_{jsh} - x_{ish}) + \sum_{s\in S}\sum_{r\in R} s(w_{jsr} - w_{isr}) \le 0, \qquad \forall i\in N,\ j\in P_i \tag{2.63}$$

$$t_i = \sum_{s\in S}\sum_{h\in H} t^h_{ih}\cdot x_{ish} + \sum_{s\in S}\sum_{r\in R} t^r_{ir}\cdot w_{isr}, \qquad \forall i\in N \tag{2.64}$$

$$t^s_i - t^s_j + M\left(1 - \sum_{h\in H}(x_{jsh} + x_{ish})\right) + M\left(1 - \sum_{r\in R}(w_{jsr} + w_{isr})\right) \ge t_j, \qquad \forall i\in N,\ s\in S,\ j\in P_i \tag{2.65}$$

$$t^s_j - t^s_i + M(1 - x_{jsh}) + M(1 - x_{ish}) + M(1 - u_{ij}) \ge t_i, \qquad \forall i\in N,\ s\in S,\ h\in H,\ j\in PF^0_i \tag{2.66}$$

$$t^s_i - t^s_j + M(1 - x_{jsh}) + M(1 - x_{ish}) + M\cdot u_{ij} \ge t_j, \qquad \forall i\in N,\ s\in S,\ h\in H,\ j\in PF^0_i \tag{2.67}$$

$$t^s_j - t^s_i + M(1 - w_{jsr}) + M(1 - w_{isr}) + M(1 - v_{ij}) \ge t_i, \qquad \forall i\in N,\ s\in S,\ r\in R,\ j\in PF^0_i \tag{2.68}$$

$$t^s_i - t^s_j + M(1 - w_{jsr}) + M(1 - w_{isr}) + M\cdot v_{ij} \ge t_j, \qquad \forall i\in N,\ s\in S,\ r\in R,\ j\in PF^0_i \tag{2.69}$$

$$\sum_{i\in N} x_{ish} - n\cdot z^h_{sh} \le 0, \qquad \forall s\in S,\ h\in H \tag{2.70}$$

$$\sum_{i\in N} w_{isr} - n\cdot z^r_{sr} \le 0, \qquad \forall s\in S,\ r\in R \tag{2.71}$$

$$\sum_{h\in H} z^h_{sh} - n^h\cdot y_s \le 0, \qquad \forall s\in S \tag{2.72}$$

$$\sum_{r\in R} z^r_{sr} - n^r\cdot y_s \le 0, \qquad \forall s\in S \tag{2.73}$$

$$y_s \ge y_{s+1}, \qquad \forall s\in S\setminus\{m\} \tag{2.74}$$

$$t^s_i + t_i \le c, \qquad \forall i\in N \tag{2.75}$$

Constraint (2.62) is a single assignment constraint, which restricts a task to be performed by a single operator at a single station. Constraint (2.63) ensures that all task precedence relations are satisfied. Constraint (2.64) determines the duration of each task after it has been assigned to a human or robot. Constraint (2.65) ensures that tasks with immediate predecessors are not started until after their immediate predecessors have finished.

If two tasks i and j have no precedence relation but are assigned to the same human or robot, either Constraint (2.65) or (2.66) for the human or Constraint (2.67) or (2.68) for the robot become active. For a human, if i is assigned earlier than j ($u_{ij} = 1$), then Constraint (2.65) becomes $t^s_j - t^s_i \ge t_i$. Otherwise, if j is assigned earlier than i, then Constraint (2.66) becomes $t^s_i - t^s_j \ge t_j$. For a robot, if task i is assigned earlier than j ($v_{ij} = 1$), then Constraint (2.67) becomes $t^s_j - t^s_i \ge t_i$. Otherwise, if j is assigned earlier than i, then Constraint (2.68) becomes $t^s_i - t^s_j \ge t_j$. With a sufficiently large number M, if two tasks are assigned to different stations, they are not considered in Constraint (2.64). Furthermore, suppose two tasks are assigned to different humans. In that case, they are not considered in Constraints (2.65) and (2.66), and if two tasks are assigned to different robots, they are not considered in Constraints (2.67) and (2.68).

Constraint (2.70) ensures that if a task is assigned to human h and that human is assigned to station s, then the variable z^h_{sh} is turned on. Similarly, Constraint (2.71) ensures that if a task is assigned to robot r and that robot is assigned to station s, then the variable z^r_{sr} must be turned on. Constraints (2.72) and (2.73) ensure the activation of the station variable y_s if at least one human or robot, respectively, is assigned to that station. Furthermore, these constraints also limit the number of humans and robots that can be assigned to each station to their respective maximums. Constraint (2.74) ensures that a station cannot be used unless the previous station is also used. Constraint (2.75) ensures that the finish time of each task satisfies the given cycle time.

2.8 HRALBP-2

The aim of the assembly line balancing problem with human-robot collaboration Type-2 (HRALBP-2) is to minimise cycle time, given that tasks may be performed by humans or robots. For HRALBP-2, define the following binary and continuous variables.

Variable	Description	Defined for
$x_{ish} \in \{0,1\}$	1 if task i is assigned to human h at station s, 0 otherwise	$i \in N,\ s \in S,\ h \in H$
$w_{isr} \in \{0,1\}$	1 if task i is assigned to robot r at station s, 0 otherwise	$i \in N,\ s \in S,\ r \in R$
$u_{ij} \in \{0,1\}$	1 if task i completed before j by the same human, 0 otherwise	$i, j \in N$
$v_{ij} \in \{0,1\}$	1 if task i completed before j by the same robot, 0 otherwise	$i, j \in N$
$z^h_{sh} \in \{0,1\}$	1 if human h is used in station s, 0 otherwise	$s \in S,\ h \in H$
$z^r_{sr} \in \{0,1\}$	1 if robot r is used in station s, 0 otherwise	$s \in S,\ r \in R$
$t_i \in \mathbb{R}$	duration of task i with its assigned operator	$i \in N$
$t^s_i \in \mathbb{R}$	start time of task i	$i \in N$
$t^f_i \in \mathbb{R}$	finish time of task i	$i \in N$
$t^a_s \in \mathbb{R}$	finish time of all tasks assigned to station s	$s \in S$
$c \in \mathbb{R}$	cycle time	

Table 2.9: Variable definitions for HRALBP-2.

HRALBP-2 is formulated as follows:

$$\min \quad c$$

s.t.

Constraints (2.62)-(2.71)

$$\sum_{h \in H} z^h_{sh} - n^h \leq 0, \qquad \forall s \in S \tag{2.76}$$

$$\sum_{r \in R} z^r_{sr} - n^r \leq 0, \qquad \forall s \in S \tag{2.77}$$

$$t^s_i + t_i \leq t^f_i, \qquad \forall i \in N \tag{2.78}$$

$$t^f_i \leq t^a_s + M\left(1 - \sum_{h \in H} x_{ish} - \sum_{r \in R} w_{isr}\right), \qquad \forall i \in N,\ s \in S \tag{2.79}$$

$$t^a_s \leq c, \qquad \forall s \in S \tag{2.80}$$

Constraints (2.76) and (2.77) ensure that each station does not exceed the maximum number of allowed humans and robots, respectively. Constraint (2.78) determines the completion time of each task, and Constraint (2.79) determines the completion time of all tasks that are assigned to each station. Constraint (2.80) connects the cycle time to the completion time of each station.

2.9 HRALBP-3

The aim of the assembly line balancing problem with human-robot collaboration Type-3 (HRALBP-3) is to minimise the cost of setting up the assembly line given a set cycle time. This type takes into account station cost, human salary, robot purchasing cost and cost of robot energy usage. As minimising station cost is one of the requirements, this is just a form of minimising the number of stations. Hence, to create a formulation for HRALBP-3, Nourmohammadi et al. (2024) created a new objective value and provided additional constraints to their HRALBP-1 formulation.

The Type-3 human-robot problem also contains additional restrictions on task assignment. Similar to TALBP-2, there are positive and negative zoning restrictions that determine tasks that can and cannot be assigned to the same station. Negative positioning restrictions, similar to predetermined position restrictions in TALBP-2, determine the stations to which tasks can be assigned. Specifically, these restrictions determine the stations to which each task cannot be assigned.

Like humans, robots require specific tools to complete certain tasks. For this problem type, we assume each robot has access to all the tools required to complete each task and can switch between them as needed. So, we also restrict the number of tool changes allowed for each robot at each station. The purpose of this is to group tasks that require the same tools together, thereby reducing inefficiency caused by frequent tool changes. While there is no time penalty associated with tool changes in this problem time, in reality, this would be the case; hence, limiting tool changes is a necessary element of the formulation.

The last restriction is joint performance. Similar to synchronism in TALBP-2, joint restrictions determine pairs of tasks that must be performed simultaneously. In the case of this problem, however, additional complexity comes from the need for a single human and a single robot to perform a joint task each.

Considering these restrictions, define the variables in Table 2.10, and the following formulation for HRALBP-3.

Variable	Description	Defined for
$x_{ish} \in \{0,1\}$	1 if task i is assigned to human h at station s, 0 otherwise	$i \in N,\ s \in S,\ h \in H$
$w_{isr} \in \{0,1\}$	1 if task i is assigned to robot r at station s, 0 otherwise	$i \in N,\ s \in S,\ r \in R$
$w^r_{ijsr} \in \{0,1\}$	1 if tasks i and j are completed at station s by robot r, 0 otherwise	$i, j \in N,\ s \in S,\ r \in R$
$u_{ij} \in \{0,1\}$	1 if task i completed before j by the same human, 0 otherwise	$i, j \in N$
$v_{ij} \in \{0,1\}$	1 if task i completed before j by the same robot, 0 otherwise	$i, j \in N$
$z^h_{sh} \in \{0,1\}$	1 if human h is used in station s, 0 otherwise	$s \in S,\ h \in H$
$z^r_{sr} \in \{0,1\}$	1 if robot r is used in station s, 0 otherwise	$s \in S,\ r \in R$
$y_s \in \{0,1\}$	1 if station s is used, 0 otherwise	$s \in S$
$t_i \in \mathbb{R}$	duration task i with its assigned operator	$i \in N$
$t^s_i \in \mathbb{R}$	start time of task i	$i \in N$

Table 2.10: Variable definitions for HRALBP-3.

$$\min \quad \sum_{s\in S}(c^s_s \cdot y_s) + \sum_{s\in S}\sum_{h\in H}\left(c^h_h \cdot z^h_{sh}\right) + \sum_{s\in S}\sum_{r\in R}(c^r_r \cdot z^r_{sr}) + \sum_{s\in S}\sum_{r\in R}\sum_{i\in N}(e_{ir} \cdot w_{isr}) \tag{2.81}$$

s.t.

Constraints (2.62) - (2.75)

$$\sum_{h\in H}\sum_{r\in R}(x_{ish} + w_{isr}) = \sum_{h\in H}\sum_{r\in R}\left(x_{jsh} + w_{jsr}\right), \qquad \forall (i,j) \in PZ,\ s \in S \tag{2.82}$$

$$\sum_{h\in H}\sum_{r\in R}(x_{ish} + w_{isr}) + \sum_{h\in H}\sum_{r\in R}\left(x_{jsh} + w_{jsr}\right) \leq 1, \qquad \forall (i,j) \in NZ,\ s \in S \tag{2.83}$$

$$\sum_{h\in H}\sum_{r\in R}(x_{ish} + w_{isr}) = 0, \qquad \forall (i,s) \in NC \tag{2.84}$$

$$t^s_i = t^s_j, \qquad \forall (i,j) \in JT \tag{2.85}$$

$$\sum_{h\in H} x_{ish} + \sum_{h\in H} x_{jsh} \leq 1, \qquad \forall (i,j) \in JT,\ s \in S \tag{2.86}$$

$$\sum_{r\in R} w_{isr} + \sum_{r\in R} w_{jsr} \leq 1, \qquad \forall (i,j) \in JT,\ s \in S \tag{2.87}$$

$$\sum_{h\in H} x_{ish} + \sum_{r\in R} w_{isr} = \sum_{h\in H} x_{jsh} + \sum_{r\in R} w_{jsr}, \qquad \forall (i,j) \in JT,\ s \in S \tag{2.88}$$

$$w^r_{ijsr} \leq w_{isr}, \qquad \forall i,j \in N,\ s \in S,\ r \in R \tag{2.89}$$

$$w^r_{ijsr} \leq w_{jsr}, \qquad \forall i,j \in N,\ s \in S,\ r \in R \tag{2.90}$$

$$w^r_{ijsr} \geq w_{isr} + w_{jsr} - 1, \qquad \forall i,j \in N,\ s \in S,\ r \in R \tag{2.91}$$

$$\sum_{(i,j)\in TC} w^r_{ijsr} \leq TC_m, \qquad \forall s \in S,\ r \in R \tag{2.92}$$

The objective function in Equation (2.81) minimises the costs of station setup, human salary, and robot purchasing and energy consumption. Constraint (2.82) ensures that pairs of tasks with positive zoning restrictions are completed at the same station, and Constraint (2.83) ensures the opposite for pairs of tasks with negative zoning restrictions. Constraint (2.84) ensures tasks are only completed in their allowed stations. Constraints (2.85) - (2.88) handle joint tasks. Constraint (2.85) ensures that tasks required to be performed by human and robot collaboration are started at the same time (similar to synchronism restrictions in Section 2.6 for TALBP-2). Constraints (2.86) and (2.87) ensure that no more than one human and one robot (respectively) are assigned to pairs of joint tasks. Constraint (2.88) ensures that one human and one robot are specifically assigned to perform joint tasks collaboratively. Constraints (2.89) - (2.91) are a linearisation of the non-linear constraint $w^r_{ijsr} = w_{isr} \cdot w_{jsr}$, which, when combined with Constraint (2.92), limits the number of tool changes for each robot at each station, while calculating the total tool changes between pairs of tasks.

2.10 Proposed Improvements

In the above TALBP-1 formulation, stations are allowed to be activated even if the station before it is deactivated. This results in a much larger solution space, as well as significant symmetry. This is because any two stations, as long as they are in the correct order, are mathematically identical to any other two stations. This creates many solutions with the same objective value. This issue, as well as alternative time limit and precedence constraints, is addressed in the following chapter.

The main issue with the above U-shaped, two-sided, and human-robot formulations is the use of Big-M constraints, as seen in Constraints (2.46) and (2.47). Big-M constraints are defined as constraints where large positive coefficients are introduced to activate/deactivate decision variables. They are well known for creating poor linear programming relaxations, and hence are likely to slow the solution time of the problem (Codato and Fischetti, 2006). Additionally, the selection of the value of the large positive number is extremely important, as too small a number can eliminate valid solutions to the problem. Hence, the formulations presented in Chapter 3 aim to eliminate many of the Big-M constraints in these formulations.

TALBP and HRALBP are scheduling problems, as they require scheduling tasks along the line in a way that satisfies precedence conditions and joint work conditions. It has been shown in the literature that constraint programming (CP) outperforms mixed-integer linear programming on many benchmark scheduling problem instances (Güner et al., 2024; Lunardi et al., 2020; Zeng et al., 2025). Hence, CP alternatives are proposed in the following chapter.

Chapter 3

New Formulations

In this chapter, new formulations are proposed for some of the problems introduced in the previous chapter. As many of these formulations will utilise constraint programming (CP), Table 3.1 outlines the notation for CP-specific variables and constraints.

Notation	Description
$t \in [a,b]$	Integer variable within the domain $[a,b]$.
$I = \text{opt}(a,b,x)$	Optional fixed sized interval variable defined by the tuple (a,b,x), where a is the start time of the interval, b is the duration of the interval, and x is the binary indicator variable that the determines if the interval is activated (if the interval is in use within the solution).
$\text{addExactlyOne}\left(\{x_d\}_{d\in D}\right)$	Adds a single assignment constraint to x over the domain D. Serves the same purpose as $\sum_{d\in D} x_d = 1$.
$\text{addAtMostOne}\left(\{x_d\}_{d\in D}\right)$	Sets at most one variable x over the domain D to be true. Serves the same purpose as $\sum_{d\in D} x_d \leq 1$.
$\text{addMinEquality}\left(x, \{y_d\}_{d\in D}\right)$	Sets $x = \min_{d\in D} y_d$. y may be another variable or some real value.
$\text{noOverlap}(I_d)_{d\in D}$	Ensures that all activated intervals in the domain D cannot have any overlap with each other, i.e., an interval cannot start until the previous interval has completed.
$\text{not}(x)$	Returns the negation of the binary variable x.

Table 3.1: Constraint programming logical constraints notation.

3.1 SALBP-1

To improve upon the models proposed in Section 2.1, define the following two additional constraints (using the variables as defined in Table 2.1):

$$\sum_{u\in[E_i,s]} x_{iu} \leq \sum_{u\in[E_j,s]} x_{ju}, \qquad \forall i \in N,\ j \in F_i,\ s \in [E_j, L_i] \tag{3.1}$$

$$\sum_{i\in W_s} t_i \cdot x_{is} \leq c \cdot y_s, \qquad \forall s \in S \tag{3.2}$$

The aim of Constraint (3.1) is to further strengthen the precedence constraints by only considering the stations where tasks i and j can be performed. Constraint (3.2) can be used to replace Constraints (2.3) and (2.5). This works by setting the "local" cycle time at each station to zero if that station is not in use, thereby disallowing any task allocation to that station. Furthermore, if the station is used, it also limits the tasks performed at that station to satisfy the cycle time requirement. Using these new constraints, define new models, M8, M9 and M10 as shown in Table 3.2.

Base Model	(2.2), (2.6), (2.8)
Model	**Constraints**
M8	(2.3), (2.5),(3.1)
M9	(2.4), (3.2)
M10	(3.1), (3.2)

Table 3.2: Additional SALBP-1 models.

3.2 TALBP-1

3.2.1 Improved MILP Formulation

To make the objective value and connections between mated-stations and stations more straightforward, redefine the variable u_s in Table 3.3 (also containing the other variables used in the new formulation).

Variable	Description	Defined for
$x_{isk} \in \{0,1\}$	1 if task i is assigned to workstation s on side k, 0 otherwise	$i \in N,\ s \in S,\ k \in K_i$
$u_s \in \{0,1\}$	1 if mated-station s is used, 0 otherwise	$s \in S$
$y_{sk} \in \{0,1\}$	1 if mated-station s is used on side k	$s \in S,\ k \in K$
$z_{ij} \in \{0,1\}$	1 if task i is assigned earlier than task j in the same station	$i \in N,\ j \in R_i$
$t_i^f \in \mathbb{R}$	finish time of task i	$i \in N$

Table 3.3: Variable definition for new TALBP-1 MILP formulation.

The updated MILP formulation for TALBP-1 is formulated with the following objective and constraints:

$$\min \quad \sum_{s\in S} u_s + \varepsilon \sum_{s\in S}\sum_{k\in K} y_{sk} \tag{3.3}$$

s.t.

$$\text{Constraints (2.42) - (2.48), (2.50)} \tag{3.4}$$

$$y_{sk} \leq u_s, \qquad \forall s \in S,\ k \in K \tag{3.5}$$

$$u_s \geq u_{s+1}, \qquad \forall s \in S \setminus \{m\} \tag{3.6}$$

$$\sum_{i\in N|k\in K_i} (t_i \cdot x_{isk}) \leq c, \qquad \forall s \in S,\ k \in K \tag{3.7}$$

The objective function (3.3) minimises the number of mated-stations as a primary objective and the number of individual stations as a secondary objective. Constraint (3.5) serves to connect the mated-station and station variables such that a mated-station cannot be used unless one of the individual stations at that mated-station is used. Constraint (3.6) acts in the same way as Constraint (2.8) for SALBP-1, in which it only allows a station to be activated if the previous station is already activated. Not only does this make the solution neater, but it also decreases the solution space and symmetry of the formulation. Constraint (3.7) serves to strengthen the formulation without impacting the solution by further enforcing the maximum cycle time on each station.

3.2.2 New CP Formulation

If we treat each side of the assembly line as a single continuous station that spans the length of the line, precedence constraints become trivial, as the only requirement is that a predecessor is completed before its follower is started. However, to track the station use (as required by Type-1 formulations), we must consider when the tasks are completed. By placing imaginary barriers along each side of the line at increments of c (given cycle time) and limiting the intervals of tasks so that they do not cross those barriers, we ensure that tasks are completed within a single station, rather than over multiple stations. Furthermore, by defining optional intervals that span individual stations, we can track which stations are used. This, rather than a direct adaptation of the MILP formulation into CP, aims to leverage the distinct advantages of constraint programming. Hence, define the variables as shown in Table 3.4.

Variable	Description	Defined for
$x_{ik} \in \{0,1\}$	1 if task i is completed on side k of the line, 0 otherwise	$i \in N,\ k \in K$
t_i^s	integer start time of task i within the total assembly line time	$i \in N$
$I_{ik} = \text{opt}\,(t_i^s,\ t_i,\ x_{ik})$	optional fixed-size interval for task i completed on side k	$i \in N,\ k \in K$
$J_{sk} = \text{opt}\,((s-1)c,\ c,\ \text{not}(y_{sk}))$	optional fixed-size interval for station s on side k not activated	$i \in N,\ k \in K$

Table 3.4: Variable definitions for TALBP-1 for CP formulation.

To prevent task intervals from crossing over station barriers, restrict the domain for the start times as follows:

$$t_i^s \in \bigcup_{s \in S} [(s-1)c,\ s \cdot c - t_i], \qquad \forall i \in N \tag{3.8}$$

This ensures that the latest a task can start within a station is the cycle time minus the task's duration. The interval variable I serves to keep track of when each task is started and completed, and the variable J serves to create an interval domain for each station on each side of the line. This interval is only enabled if the station it corresponds to is not activated. Therefore, TALBP-1 can be represented by the following CP formulation:

$$\min \quad \sum_{s\in S} u_s + \varepsilon \sum_{s\in S}\sum_{k\in K} y_{sk}$$

s.t.

Constraints (3.5), (3.6)

$$\text{addExactlyOne}\left(\{x_{ik}\}_{k\in K_i}\right), \qquad \forall i \in N \tag{3.9}$$

$$t_j^s + t_j + \leq t_i^s, \qquad \forall i \in N,\ j \in P_i \tag{3.10}$$

$$\text{noOverlap}\left(\{J_{sk}\}_{s\in S} \cup \{I_{ik}\}_{i\in N}\right), \qquad \forall k \in K \tag{3.11}$$

Constraint (3.9) ensures that a task can only be performed once, and Constraint (3.10) ensures that tasks with precedence conditions are completed in the correct order. Constraint (3.11) handles the scheduling of the formulation. If a station (s,k) is not activated, then J_{sk} will be enabled; this will then be used within the no-overlap constraint, not allowing any task intervals to be used at that station. Furthermore, this constraint also ensures that no tasks on the same side overlap in their completion. This restricts tasks to be performed one after the other, while keeping track of which stations are active.

3.3 TALBP-2

In addition to developing a new constraint programming formulation for TALBP-2, the MILP can be strengthened by using Constraint (3.7). This provides a stronger lower bound on the cycle time and hence the objective value.

3.3.1 New CP Formulation

While it would be beneficial to develop a CP formulation for TALBP-2 similar to that of TALBP-1 above, stations in Type-2 problems are not of fixed length (due to variable cycle time). In preliminary testing, this caused the formulation to be substantially weaker, where for all large test instances, it performed worse than the base MILP shown in Section 2.6. Furthermore, as positive and negative zoning constraints require the model to keep track of the station and side on which each task is performed, treating each side as a single continuous station would not allow for these constraints to be implemented. Hence, the model below is a direct adaptation of the MILP into CP, with the scheduling constraints replaced with no-overlap constraints on each station and side.

Define an additional parameter, c^t, to be the cut-off for the cycle time, calculated as shown in Equation (3.12).

$$c^t = \sum_{i\in N} t_i \tag{3.12}$$

Using this, the following variables for TALBP-2 are defined in Table 3.5.

Variable	Description	Defined for
$x_{isk} \in \{0,1\}$	1 if task i is completed at station s on side k of the line, 0 otherwise	$i \in N,\ s \in S,\ k \in K$
$t_i^s \in [0, c^t]$	integer start time of task i within cycle time	$i \in N$
$I_{isk} = \text{opt}\,(t_i^s,\ t_i,\ x_{isk})$	optional fixed-size interval for task i completed on side k	$i \in N,\ s \in S,\ k \in K$
$c \in [0, c^t]$	integer cycle time	

Table 3.5: Variable definitions for TALBP-1 for CP formulation.

Hence, the CP model for TALBP-2 can be formulated as follows:

$$\min \quad c$$

s.t.

Constraints (2.52), (2.57), (2.58), (2.60), (3.7)

$$\text{addExactlyOne}(\{x_{isk}\}_{s\in S,\ k\in K_i}), \qquad \forall i \in N \tag{3.13}$$

$$\text{addAtMostOne}(\{x_{isk}\}_{k\in K_i} \cup \{x_{jsk}\}_{k\in K_j}), \qquad \forall (i,j) \in NZ, s \in S \tag{3.14}$$

$$t_i^s = t_j^s, \qquad \forall (i,j) \in SC \tag{3.15}$$

$$t_i^s + t_i \le c, \qquad \forall i \in N \tag{3.16}$$

$$t_j^s + t_j + \sum_{s\in S}\sum_{k\in K_j}(s \cdot c^t \cdot x_{jsk}) \le t_i^s + \sum_{s\in S}\sum_{k\in K_i}(s \cdot c^t \cdot x_{isk}), \qquad \forall i \in N,\ j \in P_i \tag{3.17}$$

$$\text{noOverlap}\,(\{I_{isk}\}_{i\in N})\,, \qquad \forall s \in S,\ k \in K \tag{3.18}$$

Constraint (3.13) is a single assignment constraint, ensuring that a task is only assigned to a single station on a side it can be assigned to. Constraint (3.14) ensures that tasks with negative zoning restrictions are not assigned to the same mated-station. Constraint (3.15) ensures tasks with synchronism constraints start at the same time, and Constraint 3.16 calculates the cycle time based on the finishing time of each task. Constraint (3.17) satisfies precedence by ensuring that predecessor tasks must be completed before their followers can be started. Constraint (3.18) handles the scheduling of the solution, ensuring that no two tasks overlap each other when performed at the same station on the same side of the line.

3.4 HRALBP

In Chapter 2, HRALBP formulations made use of different variables for both operator types (humans and robots). Combining these and using an additional index to differentiate between the two operator types simplifies the formulation. Hence, additional sets and data are defined in Table 3.

As a requirement of human-robot problems is to keep track of which humans and robots are used at each station, and which humans and robots perform each task, we are required to track the stations in which each task is performed. Hence, this problem cannot be formulated in the same manner as the CP formulation for TALBP-1 presented above, in which we treat each side (or, in this case, each operator) as a single continuous station. Furthermore, since the time it takes for each task to be completed is determined by the operator it is assigned to, we cannot restrict the domain of the start times as in Equation (3.8). Hence, the following formulations for human-robot problems do not consider each operator as existing in a single continuous station, but instead as separate stations along the line. So, define the variables for HRALBP problems in Table 3.6.

Variable	Description	Defined for
$x_{islo} \in \{0,1\}$	if task i is assigned to operator l of type o at workstation s, 0 otherwise	$\forall i \in N,\ s \in S,\ o \in O,\ l \in L$
$z_{slo} \in \{0,1\}$	if operator l of type o is assigned to workstation s, 0 otherwise	$\forall s \in S,\ o \in O,\ l \in L$
$y_s \in \{0,1\}$	if station s is used (HRALBP-1 and 3)	$s \in S$
$t_i \in [0, c^t]$	integer time to complete task i with its assigned operator	$i \in N$
$t_i^s \in [0, c^t]$	integer start time of task i within cycle time	$i \in N$
$I_{islo} = \text{opt}\left(t_i^s,\ t_{ilo}^l,\ x_{islo}\right)$	optional fixed-size interval for task i completed on by operation l of type o at station s	$i \in N,\ s \in S,\ o \in O,\ l \in L$
$c \in [0, c^t]$	integer cycle time (HRALBP-2)	
$w_{ijsl}^r \in \{0,1\}$	1 if tasks i and j are completed at station s by robot l, 0 otherwise (HRALBP-3)	$(i,j) \in JT,\ s \in S,\ l \in L$

Table 3.6: Variable definitions for HRALBP CP formulations.

For HRALBP-1 and HRALBP-3, the cycle time cut-off is set to the maximum cycle time, i.e., $c^t = c$. For HRALBP-2, c^t is calculated using the following equation:

$$c^t = \sum_{i \in N} \left(\max_{o \in O,\ l \in L} \left\{ t_{ilo}^l \right\} \right)$$

3.4.1 HRALBP-1

HRALBP-1 is formulated as follows using constraint programming:

$$\min \quad \sum_{s \in S} y_s$$

s.t.

Constraints (2.74), (2.75)

$$\text{addExactlyOne}(\{x_{islo}\}_{s \in S,\ o \in O,\ l \in L}), \qquad \forall i \in N \tag{3.19}$$

$$\sum_{s \in S} \sum_{l \in L} \sum_{o \in O} (s \cdot x_{jslo}) \leq \sum_{s \in S} \sum_{l \in L} \sum_{o \in O} (s \cdot x_{islo}), \qquad \forall i \in N,\ j \in P_i \tag{3.20}$$

$$t_i = \sum_{s\in S}\sum_{l\in L}\sum_{o\in O}(t^l_{ilo}\cdot x_{islo}), \qquad \forall i \in N \tag{3.21}$$

$$t^s_i - t^s_j + c\left(1-\sum_{o\in O}\sum_{l\in L}x_{jslo}\right) + c\left(1-\sum_{o\in O}\sum_{l\in L}x_{islo}\right) \geq t_j, \qquad \forall i \in N,\ s\in S,\ j\in P_i \tag{3.22}$$

$$\sum_{i\in N}x_{islo} - n\cdot z_{slo} \leq 0, \qquad \forall s\in S,\ o\in O,\ l\in L \tag{3.23}$$

$$\sum_{l\in L}z_{slo} - n^l_o\cdot y_s \leq 0, \qquad \forall s\in S,\ o\in O \tag{3.24}$$

$$\text{noOverlap}\left(\{I_{islo}\}_{i\in N}\right), \qquad \forall s\in S,\ o\in O,\ l\in L \tag{3.25}$$

Constraint (3.19) is a single assignment constraint, ensuring that each task is only assigned to a single station and operator. Constraint (3.20) handles precedence by restricting tasks to be done either in the same station or in a station after their predecessors. Constraint (3.21) determines the duration of each task once it has been assigned to an operator. Constraint (3.22) handles precedence within the same station, ensuring predecessors are completed before their followers can be started. Constraint (3.23) connects the worker assignment variable z to the task assignment variable x, to track which operator completes each task and at which station. Constraint (3.24) limits the number of each operator type allowed at each station, setting this maximum to zero if the station is inactive. Constraint (3.25) handles the scheduling of the formulation, ensuring that no two tasks overlap each other when being performed by the same operator at the same station.

3.4.2 HRALBP-2

In the original HRALBP-2 MILP formulation, an additional Big-M constraint (Constraint (2.79)) is used to determine the time when all tasks in each station are completed. However, not only does this weaken the formulation, but it is also unnecessary to determine this value. It is only required to minimise the time that the last task at each station finishes. The largest of these last finishing times is then the cycle time for the whole line. Therefore, Constraint (2.75) suffices for this purpose. So, HRALBP-2 can be formulated in CP as follows:

$$\min \quad c$$

s.t.

Constraints (2.75), (3.19) - (3.21), (3.23), (3.25)

$$\sum_{l\in L}z_{slo} - n^l_o \leq 0, \qquad \forall s\in S,\ o\in O \tag{3.26}$$

Constraint (3.26) limits the number of operators of each type that can be assigned to each station. This serves the same purpose as Constraint (3.24) from the above HRALBP-1 formulation, if we assume all stations are activated, as is the case for Type-2 formulations.

3.4.3 HRALBP-3

The following CP formulation is largely an adaptation of the MILP model proposed in Section 2.9 and the CP formulation for HRALBP-1 proposed in Section 3.4.1, incorporating additional alternative constraints that leverage the strengths of CP.

$$\min \quad \sum_{s\in S}(c^s_s \cdot y_s) + \sum_{s\in S}\sum_{l\in L}\sum_{o\in O}\left(c^l_{lo}\cdot z_{slo}\right) + \sum_{i\in N}\sum_{s\in S}\sum_{l\in L}(e_{il}\cdot x_{isl2}) \tag{3.27}$$

s.t.

Constraints (2.74), (2.75), (2.85), (2.92), (3.19) - (3.25)

$$\sum_{l\in L}\sum_{o\in O} x_{islo} = \sum_{l\in L}\sum_{o\in O} x_{jslo}, \qquad \forall (i,j)\in PZ,\ s\in S \tag{3.28}$$

$$\text{addAtMostOne}\left(\{x_{islo}\}_{l\in L,\ o\in O} \cup \{x_{jslo}\}_{l\in L,\ o\in O}\right), \qquad \forall (i,j)\in NZ,\ s\in S \tag{3.29}$$

$$\sum_{l\in L}\sum_{o\in O} x_{islo} = 0, \qquad \forall (i,s)\in NC \tag{3.30}$$

$$\text{addAtMostOne}\left(\{x_{islo}\}_{l\in L} \cup \{x_{jslo}\}_{l\in L}\right), \qquad \forall (i,j)\in JT,\ s\in S,\ o\in O \tag{3.31}$$

$$\sum_{l\in L}\sum_{o\in O} x_{islo} = \sum_{l\in L}\sum_{o\in O} x_{jslo}, \qquad \forall (i,j)\in JT,\ s\in S \tag{3.32}$$

$$\text{addMinEquality}(w^r_{ijsl}, \{x_{isl2}, x_{jsl2}\}), \qquad \forall (i,j)\in JT,\ s\in S \tag{3.33}$$

Equation (3.27) contains the new objective function, minimising station setup cost, operator setup cost, and robot energy cost. Constraints (3.28) and (3.29) respectively ensure that positive and negative zoning restrictions are satisfied. Constraint (3.30) restricts tasks to be performed only in stations in which they are allowed. Constraint (3.31) ensures that no more than one operator of each type is assigned to pairs of joint tasks. Constraint (3.32) ensures that pairs of joint tasks are performed at the same station. Constraint (3.33) calculates $w^r_{ijsr} = w_{isr}\cdot w_{jsr}$, which is used to track the number of tool changes for each robot at each station. Since both variables are binary, if either of them is not active (i.e. zero), the multiplication will be equal to zero; hence, using a minimum in this case is equivalent to multiplication.

Chapter 4

Computational Experiments

In this chapter, I test the state-of-the-art formulations presented in Chapter 2 and compare their results with those of the new formulations presented in Chapter 3. In the results presented below, MILP or MILP-1 refers to the models presented in Chapter 2, and MILP-2 and CP refer to the MILP and CP models, respectively, presented in Chapter 3.

All computational experiments were performed on an Intel® Core™ i7-13700H processor (14 Cores, 20 Threads), 16GB LPDDR5 4800 MT/s memory, and Windows 11 Home operating system. All models were implemented in Python 3.13.5, with MILP models implemented using the Gurobi 12.0.3 Python package (with a target optimality gap of zero) and CP models using the Google OR-Tools 9.14.6206 CP-SAT Solver Python package. Both solvers were given a time limit of one hour (3600 seconds) per instance.

4.1 Computational Data

For all problem types apart from TALBP, the precedence graphs were obtained from the dataset provided in Nourmohammadi et al. (2024). This dataset is an extension of standard precedence graphs commonly used in ALBP research. Namely Bowman ($n = 8$), Jackson ($n = 11$), Rozieg ($n = 25$), Buxey ($n = 29$), Sawyer ($n = 30$), Gunther ($n = 35$), Kilbridge ($n = 45$), Hahn ($n = 53$), Tonge ($n = 70$), Arcus1 ($n = 83$) and Arcus2 ($n = 111$) as compiled in Scholl (1997). Nourmohammadi et al. (2024) also provided their own Case Study precedence graph ($n = 28$). They extended the data of these precedence graphs by adding the required data for human-robot problems, such as the duration of each task for each operator. Precedence graphs specifically designed for two-sided problems were also used. P9, P12 and P24 from Kim et al. (2000), and P16, P65, P148 and P205 from Lee et al. (2001). Table 4.1 shows the number of instances tested for each problem type.

Problem Type	# Instances
SALBP-1	100
TALBP-1	35
TALBP-2	18
HRALBP-1	96
HRALBP-2	96
HRALBP-3	48

Table 4.1: Number of benchmark instances for each problem type.

4.2 SALBP-1

In testing their models on benchmark instances, Pastor et al. (2011) identified M6 and M7 to have the strongest performance. So, these two models are compared with the three proposed in Section 3.1. Table 4.2 shows the results of the computational experiments. # optimal shows the number of instances for which each formulation solved to optimality within the time limit, # feasible shows the number of instances that were solved to feasibility and # no solution shows the number of instances for which no solution was found within the time limit. avg. cpu shows the average solve time for each model and avg. gap shows the average optimality gap.

M8 and M10 solved the most instances to optimality (98 each); however, M10 is likely a superior formulation, as it had a shorter average solve time of 93.724 seconds, as opposed to 96.543 seconds for M8. Both M8 and M10 had an average optimality gap of 11.212%. These models both had the modified precedence constraint (3.1); however, M10 also had the modified time limit constraint (3.2). This suggests that the combination of these two constraints significantly strengthened the formulation, especially when compared to M9, which only had the modified time limit constraint.

	M6	M7	M8	M9	M10
# optimal	96	95	98	92	98
# feasible	4	5	2	8	2
# no solution	0	0	0	0	0
avg. cpu (s)	165.818	189.345	96.543	302.122	93.724
avg. gap (%)	11.015	11.669	11.212	11.034	11.212

Table 4.2: SALBP-1 performance comparison. avg. cpu is the mean computation time for all instances, whereas avg. gap is the mean optimality gap, considering only instances where the solver reached feasibility but not optimality.

In Figure 4.1, observe that for small instances, i.e. those that are solved in less than 100 seconds (approximately), M8 and M10 have the worst performance. But for larger instances, these formulations have the strongest performance. Furthermore, for very small instances solved in under 10 seconds, M9 has the strongest performance. Therefore, we can conclude that for large instances, the two modified constraints provide a significantly stronger formulation. In contrast, for smaller instances, the modified time limit constraint is sufficient to strengthen the formulation.

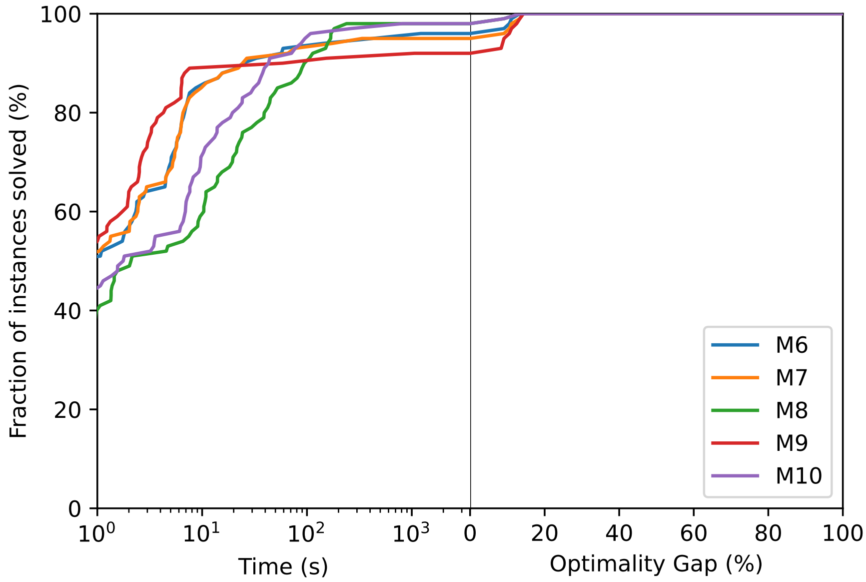

Figure 4.1: Performance plot for SALBP-1.

For complete results for SALBP-1 instances, refer to Table A.1.

4.3 TALBP-1

For the blended objective weighting, we used $\varepsilon = \frac{1}{2n+1}$, which is within the range specified in Equation (2.40). Experimenting with different values for ε within the range yielded inconsistent results. Setting it too low often resulted in differing objective values; hence, the value above was chosen. Figure 4.2 demonstrates the performance benefits of the CP formulation, as it was able to solve all instances to optimality.

Furthermore, Table 4.3 shows the performance numbers for each model. MILP-2 provided a significant performance increase over MILP-1, solving 18 instances to optimality compared to 10 for MILP-1. However, neither of the MILP formulations were able to find a feasible solution for any P148 or P205 instances. For MILP-1, this was because the solver ran out of memory when computing the solutions for P148 and P205 instances. For MILP-2, it ran out of time before finding a feasible solution for P148, and then ran out of memory when solving P205 instances.

	MILP-1	MILP-2	CP
# optimal	10	18	35
# feasible	9	1	0
# no solution	16	16	0
avg. cpu (s)	2589.45	1783.022	35.553

Table 4.3: TALBP-1 performance comparison.

Table A.2 shows the objective values and computation times for each of the TALBP-1 formulations. In some cases, MILP-1 solved to feasibility with a large optimality gap, but still returned the same

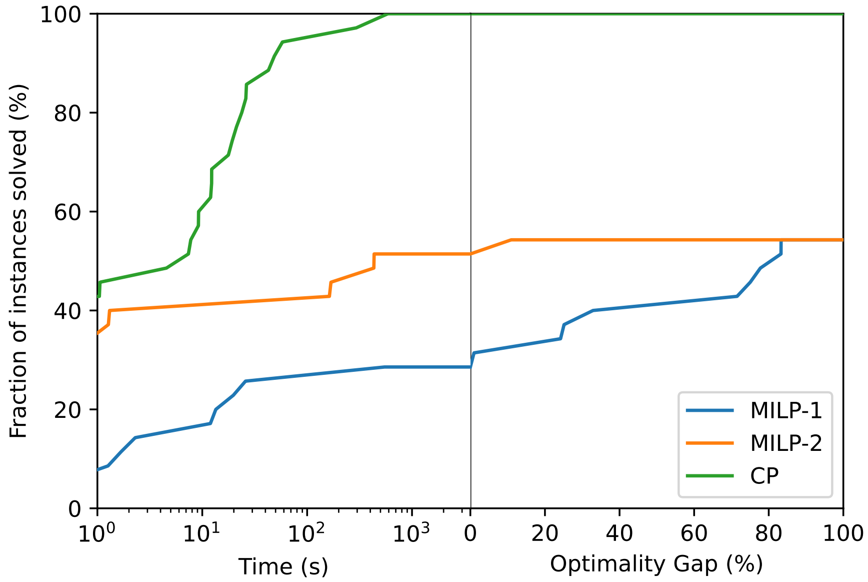

Figure 4.2: Performance plot for TALBP-1.

objective value as the other two formulations. This suggests that the solver struggled to raise the lower bound and prove that the objective value it found was optimal. The strengthening constraints introduced for MILP-2 appear to have significantly improved this. The P65 instance with $c = 490$ demonstrates this fact. Observing the solution logs (Appendices B.1 and B.2), MILP-1 reached an optimality gap of 83.3% after 160 seconds and could not close that gap before the one-hour time limit. Whereas MILP-2 reached a gap of 0.20% after 149 seconds, and a gap of 0.07% after 168 seconds. Hence, MILP-2 is a much stronger formulation than MILP-1. However, the CP formulation is superior to both MILP models because it not only solved every instance to optimality but also found a solution in the shortest time in cases where all three models could solve to optimality.

4.4 TALBP-2

Table 4.4 shows the objective values and computation times for each of the TALBP-2 formulations presented in this thesis. In cases where all three models were able to find optimal solutions, they all found the same objective value. However, for P65 and larger instances, MILP-2 was only able to solve a single instance to optimality, while MILP-1 was unable to solve any to optimality. In cases where MILP-1 and MILP-2 only found feasible solutions, MILP-2 achieved a significantly smaller optimality gap than MILP-1, demonstrating the benefit of the strengthening constraint.

Additionally, neither of the MILP formulations were able to obtain feasible solutions for any of the P205 instances, whereas the CP model did find feasible solutions within a 2.8% optimality gap. Out of the 18 benchmark instances, the CP formulation obtained optimal solutions for 14 instances and feasible solutions for four instances. Furthermore, where all three formulations achieved optimality, the CP formulation had the fastest solution time.

Problem	n	m	MILP-1		MILP-2		CP	
			Obj.	CPU (s)	Obj.	CPU (s)	Obj.	CPU (s)
P9	9	2	5.0	0.063	5.0	0.069	5.0	0.038
P9	9	3	4.0	0.056	4.0	0.082	4.0	0.023
P12	12	2	8.0	0.046	8.0	0.123	8.0	0.017
P12	12	3	5.0	0.062	5.0	0.089	5.0	0.022
P16	16	2	26.0	0.131	26.0	0.159	26.0	0.017
P16	16	3	18.0	0.211	18.0	0.461	18.0	0.043
P24	24	2	36.0	7.553	36.0	0.376	36.0	0.04
P24	24	3	25.0	1.581	25.0	0.582	25.0	0.052
P24	24	4	18.0	1.156	18.0	0.928	18.0	0.077
P65	65	4	655.0	(38.321%)	652.0	(2.147%)	647.0	(0.927%)
P65	65	6	440.0	(37.045%)	435.0	(2.299%)	431.0	458.969
P65	65	8	331.0	(17.825%)	327.0	(2.446%)	324.0	1059.347
P148	148	5	532.0	(46.429%)	517.0	(0.774%)	513.0	18.596
P148	148	7	385.0	(27.792%)	368.0	(0.543%)	366.0	162.622
P148	148	9	296.0	(3.716%)	285.0	2190.686	285.0	107.013
P205	205	6	-	-	-	-	2145.0	(2.797%)
P205	205	8	-	-	-	-	1533.0	(2.674%)
P205	205	10	-	-	-	-	1194.0	(1.591%)

Table 4.4: Benchmark results for TALBP-2.

Figure 4.3 further demonstrates the benefit of the CP formulation, as the fraction of instances solved for all computation times and optimality gaps is larger than both the MILP formulations. This implies it is more powerful for solving all instance sizes.

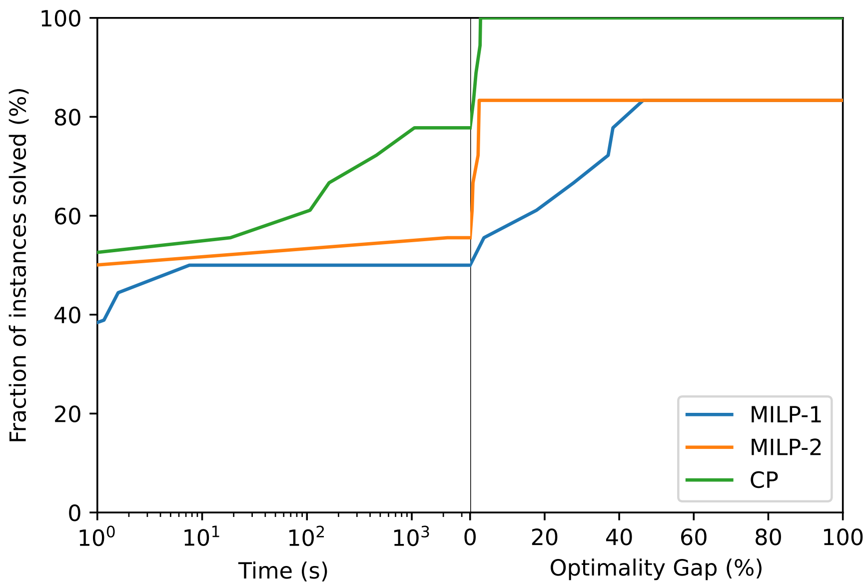

Figure 4.3: Performance plot for TALBP-2.

4.5 HRALBP-1

Figure 4.4 shows the performance of both formulations for HRALBP-1. The CP model solved more instances for each time and optimality gap, demonstrating it as the superior formulation. Additionally, Table 4.5 shows that CP solved all but four instances to optimality, with the remaining instances solved to feasibility. However, the MILP only solved 73 instances to optimality, 16 to feasibility and found no solution for seven instances. This further supports that the CP formulation is stronger. For complete results for HRALBP-1, refer to Table A.3.

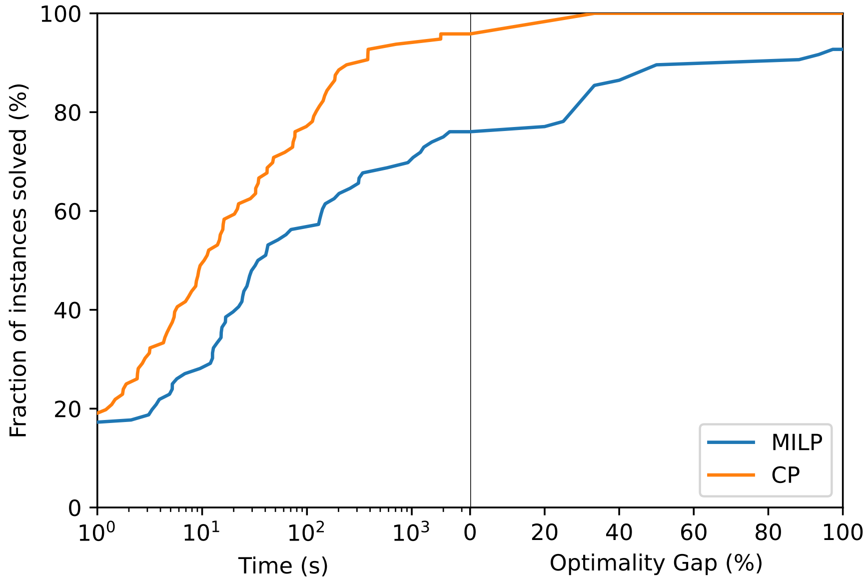

Figure 4.4: Performance plot for HRALBP-1.

	MILP	CP
# optimal	73	92
# feasible	16	4
# no solution	7	0
avg. cpu (s)	1008.3	240.441

Table 4.5: HRALBP-1 performance comparison.

For large instances ($n \geq 53$), there are solutions from both the CP and MILP formulations that do not match those in Nourmohammadi et al. (2024). It is unknown what is causing this to happen, as the MILP and CP formulations in most of these cases return different objective values.

Note that for five small instances (Bowman), the MILP formulation has better computation times. However, the computation times of the CP formulation for these instances are still less than 0.15 seconds. So, this is unimportant, as the solution time is still very fast, and the difference between the MILP and CP times for these examples is extremely small. Additionally, four Hahn instances are solved faster by the MILP model. However, since some of these have different objective values for the MILP and CP solutions, we cannot confirm that the model actually performed better in these cases.

4.6 HRALBP-2

Figure 4.5 shows the performance of the MILP and CP models for HRALBP-2. As the number of instances solved for each time and optimality gap is higher for the CP formulation, it is evident that the CP performed better overall. This is further supported by Table 4.6, as the CP was able to solve 80 of the 96 instances to optimality, whereas the MILP was able to solve only 47. For the instances not solved to optimality, both models achieved feasible solutions for the remaining instances, with MILP having an average optimality gap of 28.578% compared to 7.394% for CP. For complete results for HRALBP-2, refer to Table A.4.

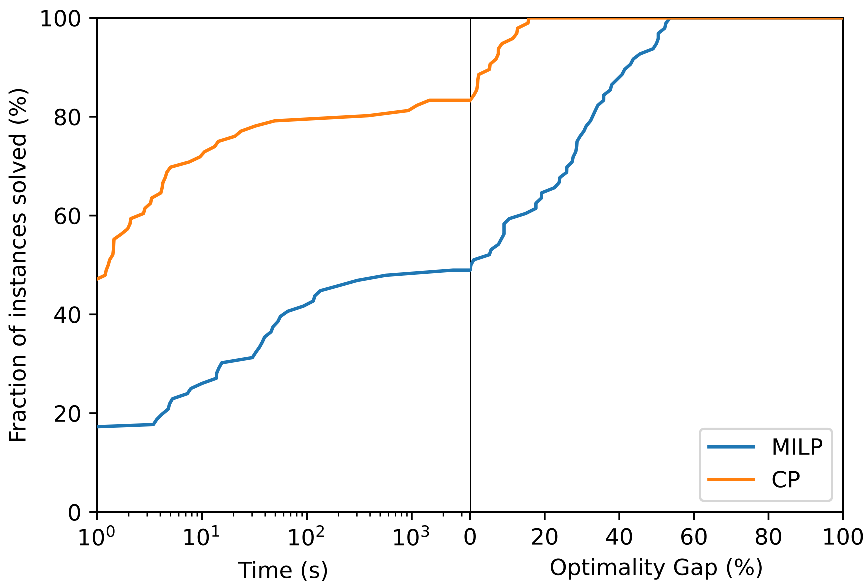

Figure 4.5: Performance plot for HRALBP-2.

	MILP	CP
# optimal	47	80
# feasible	49	16
# no solution	0	0
avg. cpu (s)	1885.351	643.479
avg. gap (%)	28.578	7.394

Table 4.6: HRALBP-2 performance comparison.

As with HRALBP-1, five small instances (Bowman) were solved faster by the MILP model than the CP. As before, the computation times for both are still extremely small, and the difference between them is negligible.

4.7 HRALBP-3

Figure 4.6 shows the performance of the MILP and CP formulations for HRALBP-3. The CP formulation has superior performance overall, with it achieving a higher number of instances solved for each time and optimality gap.

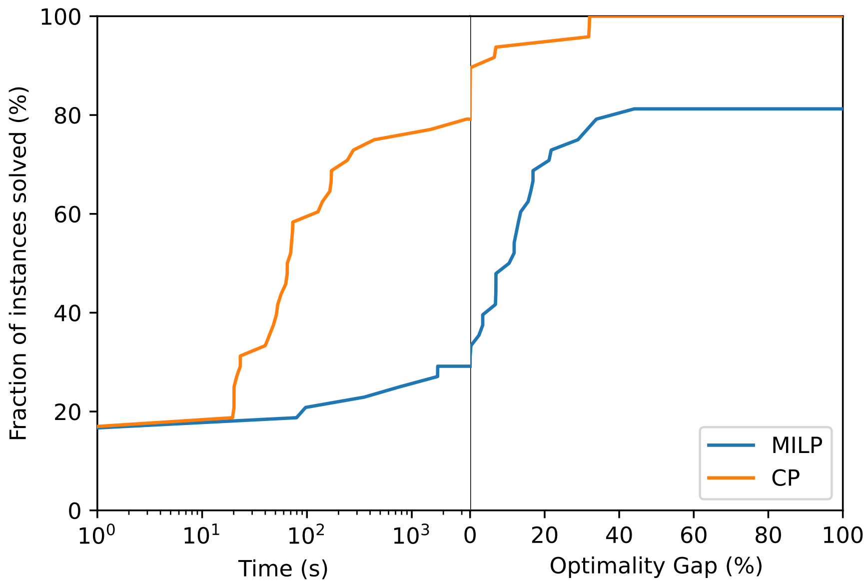

Figure 4.6: Performance plot for HRALBP-3.

	MILP	CP
# optimal	14	38
# feasible	25	10
# no solution	9	0
avg. cpu (s)	2650.993	921.433

Table 4.7: HRALBP-3 performance comparison.

Furthermore, Table 4.7 shows that the CP was able to solve 38 of the 48 instances to optimality and the other 10 to feasibility. This, compared to 14 optimal and 25 feasible for the MILP, demonstrates the strength of the CP. Additionally, the MILP was unable to find a solution for nine instances.

Observing the results for each instance in Table A.5, there are some instances where the MILP solved faster than the CP. As with the previous HRALBP problems, these instances are still solved in a very short time, making the benefits negligible compared to the time savings for larger instances.

In some cases where the MILP achieved feasibility with large optimality gaps, it still found the same objective value as the CP. This suggests the solver was able to find the objective value but struggled to raise the lower bound. We can confirm this with the solution log for the fourth Case Study instance (Appendix B.3). The optimal objective was found after 55 seconds, but the solver could not achieve a narrower gap than 28.93% within one hour.

Chapter 5

Conclusion

This thesis aimed to evaluate the effectiveness of constraint programming as an alternative to mixed-integer linear programming for solving several variants of the Assembly Line Balancing Problem (ALBP). Through comparison across six problem variants: SALBP-1, TALBP-1, TALBP-2, HRALBP-1, HRALBP-2 and HRALBP-3, the study found that CP consistently achieved faster computation times and lower optimality gaps. The proposed alternative MILP formulations also strengthened existing models but were outperformed by CP, particularly on larger benchmark instances.

Beyond comparative analysis, this thesis contributes a unified framework of exact formulations across multiple ALBP variants. The new formulations reduce reliance on Big-M constraints, improving model strength and solution performance. This work provides the foundation for further exploration of exact programming techniques in assembly line balancing research.

Işık and Yildiz (2024) compare the solution quality and time of MILP and CP, demonstrating the benefits of constraint programming in non-scheduling problems. As such, further research could be aimed towards improving non-scheduling ALBPs, such as SALBP and UALBP, through the use of constraint programming.

Recent developments in Domain-Independent Dynamic Programming (DIDP) have been applied to the study of assembly line balancing problems. In Zhang and Beck (2024), the authors implemented state-of-the-art MILP formulations and compare their performance with their own novel CP and DIDP models. The DIDP model exhibited the strongest performance for solving a set of benchmark instances, posing an additional path of investigation for improving the models discussed above.

Kucukkoc et al. (2018) adapted the formulation of the two-sided assembly line to also consider stations below the assembly line. This is essentially a three-sided assembly line. The proposed CP formulations for TALBP-1 and TALBP-2 in Chapter 3 could easily be adapted to consider a third “side” of the line, presenting additional options for future research.

As human-robot problems consider multiple workers and robots working on the same side of a line, these problems could be further expanded upon by considering workers and robots on multiple sides of the line, as in TALBP. This would further complicate the problem as it would introduce additional potential task overlaps.

In conclusion, this thesis demonstrates that constraint programming is a powerful exact solution method for modern assembly line balancing problems. By offering improved scalability, it represents a promising direction for both academic research and industrial applications in automated and human-robot collaborative manufacturing systems.

Bibliography

Mingming Zheng, Jin-Kao Hao, and Qinghua Wu. Exact and heuristic solution approaches for the generalized independent set problem. *Computers & Operations Research*, 164:106561, 2024. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2024.106561. URL `https://www.sciencedirect.com/science/article/pii/S0305054824000339`.

Rehman Zafar, Anzar Mahmood, Sohail Razzaq, Wamiq Ali, Usman Naeem, and Khurram Shehzad. Prosumer based energy management and sharing in smart grid. *Renewable and Sustainable Energy Reviews*, 82:1675–1684, 2018. ISSN 1364-0321. doi: https://doi.org/10.1016/j.rser.2017.07.018. URL `https://www.sciencedirect.com/science/article/pii/S1364032117310894`.

Reza Fachrizal, Mahmoud Shepero, Dennis van der Meer, Joakim Munkhammar, and Joakim Widén. Smart charging of electric vehicles considering photovoltaic power production and electricity consumption: A review. *eTransportation*, 4:100056, 2020. ISSN 2590-1168. doi: https://doi.org/10.1016/j.etran.2020.100056. URL `https://www.sciencedirect.com/science/article/pii/S2590116820300138`.

Haoshan Ren and Yongjun Sun. Chapter 3 - building energy flexibility: modeling and optimization. In Zhenjun Ma, Müslüm Arıcı, and Amin Shahsavar, editors, *Building Energy Flexibility and Demand Management*, pages 41–62. Academic Press, 2023. ISBN 978-0-323-99588-7. doi: https://doi.org/10.1016/B978-0-323-99588-7.00002-X. URL `https://www.sciencedirect.com/science/article/pii/B978032399588700002X`.

Francesca Rossi, Peter van Beek, and Toby Walsh. Chapter 1 - introduction. In Francesca Rossi, Peter van Beek, and Toby Walsh, editors, *Handbook of Constraint Programming*, volume 2 of *Foundations of Artificial Intelligence*, pages 3–12. Elsevier, 2006. doi: https://doi.org/10.1016/S1574-6526(06)80005-2. URL `https://www.sciencedirect.com/science/article/pii/S1574652606800052`.

Aurélien Froger, Michel Gendreau, Jorge E. Mendoza, Éric Pinson, and Louis-Martin Rousseau. Maintenance scheduling in the electricity industry: A literature review. *European Journal of Operational Research*, 251(3):695–706, 2016. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2015.08.045. URL `https://www.sciencedirect.com/science/article/pii/S0377221715008012`.

Nils Boysen, Philipp Schulze, and Armin Scholl. Assembly line balancing: What happened in the last fifteen years? *European Journal of Operational Research*, 301(3):797–814, 2022. ISSN 0377-2217. doi: 10.1016/j.ejor.2021.11.043. URL https://www.sciencedirect.com/science/article/pii/S0377221721009942.

Eduardo Álvarez Miranda and Jordi Pereira. On the complexity of assembly line balancing problems. *Computers & Operations Research*, 108:182–186, 2019. ISSN 0305-0548. doi: 10.1016/j.cor.2019.04.005. URL https://www.sciencedirect.com/science/article/pii/S0305054819300826.

Rafael Pastor, Alberto Garcia-Villoria, and Albert Corominas. Comparing ways of breaking symmetries in mathematical models for salbp-1. *Assembly Automation*, 31(4):385–387, 2011. ISSN 0144-5154. doi: 10.1108/01445151111172970. URL https://www.emerald.com/insight/content/doi/10.1108/01445151111172970/full/html.

François Margot. *Symmetry in Integer Linear Programming*, chapter 17, pages 647–686. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-540-68279-0. doi: 10.1007/978-3-540-68279-0_17. URL https://doi.org/10.1007/978-3-540-68279-0_17.

Marcus Ritt and Alysson M. Costa. Improved integer programming models for simple assembly line balancing and related problems. *International Transactions in Operational Research*, 25(4):1345–1359, 2018. doi: 10.1111/itor.12206. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/itor.12206.

Ahmed Ghoniem and Hanif D. Sherali. Defeating symmetry in combinatorial optimization via objective perturbations and hierarchical constraints. *IIE Transactions*, 43(8):575–588, 2011. doi: 10.1080/0740817X.2010.541899. URL https://doi.org/10.1080/0740817X.2010.541899.

Geoffrey Chu and Peter J. Stuckey. Dominance breaking constraints. *Constraints*, 20(2):155–182, 2015. doi: 10.1007/s10601-014-9173-7. URL https://doi.org/10.1007/s10601-014-9173-7.

Ming Li, Qiuhua Tang, Qiaoxian Zheng, Xuhui Xia, and C.A. Floudas. Rules-based heuristic approach for the u-shaped assembly line balancing problem. *Applied Mathematical Modelling*, 48:423–439, 2017. ISSN 0307-904X. doi: 10.1016/j.apm.2016.12.031. URL https://www.sciencedirect.com/science/article/pii/S0307904X17300033.

Uğur Özcan. Balancing stochastic two-sided assembly lines: A chance-constrained, piecewise-linear, mixed integer program and a simulated annealing algorithm. *European Journal of Operational Research*, 205(1):81–97, 2010. ISSN 0377-2217. doi: 10.1016/j.ejor.2009.11.033. URL https://www.sciencedirect.com/science/article/pii/S0377221709009060.

Zixiang Li, Qiuhua Tang, and Liping Zhang. Minimizing the cycle time in two-sided assembly lines with assignment restrictions: Improvements and a simple algorithm. *Mathemati-*

cal Problems in Engineering, 2016(1):4536426, 2016. doi: 10.1155/2016/4536426. URL `https://onlinelibrary.wiley.com/doi/abs/10.1155/2016/4536426`.

Amir Nourmohammadi, Masood Fathi, and Amos H.C. Ng. Balancing and scheduling human-robot collaborated assembly lines with layout and objective consideration. *Computers & Industrial Engineering*, 187:109775, 2024. ISSN 0360-8352. doi: 10.1016/j.cie.2023.109775. URL `https://www.sciencedirect.com/science/article/pii/S0360835223007994`.

Gianni Codato and Matteo Fischetti. Combinatorial benders' cuts for mixed-integer linear programming. *Operations Research*, 54(4):756–766, 2006. doi: 10.1287/opre.1060.0286. URL `https://doi.org/10.1287/opre.1060.0286`.

Funda Güner, Abdül K. Görür, Benhür Satır, Levent Kandiller, and John. H. Drake and. A constraint programming approach to a real-world workforce scheduling problem for multi-manned assembly lines with sequence-dependent setup times. *International Journal of Production Research*, 62(9): 3212–3229, 2024. doi: 10.1080/00207543.2023.2226772. URL `https://doi.org/10.1080/00207543.2023.2226772`.

Willian T. Lunardi, Ernesto G. Birgin, Philippe Laborie, Débora P. Ronconi, and Holger Voos. Mixed integer linear programming and constraint programming models for the online printing shop scheduling problem. *Computers & Operations Research*, 123:105020, 2020. ISSN 0305-0548. doi: 10.1016/j.cor.2020.105020. URL `https://www.sciencedirect.com/science/article/pii/S0305054820301374`.

Chuangfeng Zeng, Jianjun Liu, and Qinsong Li. A constraint programming approach for resource-constrained flexible assembly flow shop scheduling problem with batch direct delivery. *Computers & Operations Research*, 173:106855, 2025. ISSN 0305-0548. doi: 10.1016/j.cor.2024.106855. URL `https://www.sciencedirect.com/science/article/pii/S0305054824003277`.

Armin Scholl. Data of assembly line balancing problems. Publications of Darmstadt Technical University, Institute for Business Studies (BWL) 25075, Darmstadt Technical University, Department of Business Administration, Economics and Law, Institute for Business Studies (BWL), 1997. URL `https://assembly-line-balancing.de/wp-content/uploads/2017/01/Scholl-1993-ALBData.pdf`.

Yeo Keun Kim, Yeongho Kim, and Yong Ju Kim. Two-sided assembly line balancing: A genetic algorithm approach. *Production Planning & Control*, 11(1):44–53, 2000. doi: 10.1080/095372800232478. URL `https://doi.org/10.1080/095372800232478`.

Tae Ok Lee, Yeongho Kim, and Yeo Keun Kim. Two-sided assembly line balancing to maximize work relatedness and slackness. *Computers & Industrial Engineering*, 40(3):273–292, 2001. ISSN 0360-8352. doi: 10.1016/S0360-8352(01)00029-8. URL `https://www.sciencedirect.com/science/article/pii/S0360835201000298`.

Eyüp Ensar Işık and Seyda Topaloglu Yildiz. Integer and constraint programming models for the straight and u-shaped assembly line balancing with hierarchical worker assignment problem. *International Journal of Production Research*, 62(14):5269–5292, 2024. doi: 10.1080/00207543.2023.2290699. URL `https://doi.org/10.1080/00207543.2023.2290699`.

Jiachen Zhang and J. Christopher Beck. Domain-independent dynamic programming and constraint programming approaches for assembly line balancing problems with setups. *INFORMS Journal on Computing*, 0(0), 2024. ISSN 0305-0548. doi: 10.1287/ijoc.2024.0603. URL `https://pubsonline.informs.org/doi/10.1287/ijoc.2024.0603`.

Ibrahim Kucukkoc, Zixiang Li, Aslan D. Karaoglan, and David Z. Zhang. Balancing of mixed-model two-sided assembly lines with underground workstations: A mathematical model and ant colony optimization algorithm. *International Journal of Production Economics*, 205:228–243, 2018. ISSN 0925-5273. doi: https://doi.org/10.1016/j.ijpe.2018.08.009. URL `https://www.sciencedirect.com/science/article/pii/S0925527318303256`.

Appendix A

Complete Results Tables

A.1 SALBP-1

Problem	n	c	M6		M7		M8		M9		M10	
			Obj.	CPU (s)	Obj.	CPU (s)	Obj.	CPU (s)	Obj.	CPU (s)	Obj.	CPU (s)
Bowman	8	20	5.0	0.017	5.0	0.105	5.0	0.084	5.0	0.016	5.0	0.164
Jackson	11	7	8.0	0.104	8.0	0.134	8.0	0.112	8.0	0.083	8.0	0.133
Jackson	11	9	6.0	0.096	6.0	0.037	6.0	0.11	6.0	0.017	6.0	0.066
Jackson	11	10	5.0	0.015	5.0	0.149	5.0	0.071	5.0	0.083	5.0	0.1
Jackson	11	13	4.0	0.055	4.0	0.071	4.0	0.174	4.0	0.077	4.0	0.086
Jackson	11	14	4.0	0.066	4.0	0.066	4.0	0.118	4.0	0.106	4.0	0.1
Jackson	11	21	3.0	0.031	3.0	0.05	3.0	0.078	3.0	0.065	3.0	0.084
Roszieg	25	14	10.0	0.093	10.0	0.15	10.0	0.341	10.0	0.183	10.0	0.416
Roszieg	25	16	8.0	0.079	8.0	0.191	8.0	0.277	8.0	0.05	8.0	0.311
Roszieg	25	18	8.0	0.079	8.0	0.091	8.0	0.245	8.0	0.15	8.0	0.259
Roszieg	25	21	6.0	0.136	6.0	0.097	6.0	0.282	6.0	0.116	6.0	0.315
Roszieg	25	25	6.0	0.07	6.0	0.033	6.0	0.273	6.0	0.049	6.0	0.238
Roszieg	25	32	4.0	0.044	4.0	0.151	4.0	0.228	4.0	0.049	4.0	0.151
Buxey	29	27	13.0	0.208	13.0	0.328	13.0	0.401	13.0	0.313	13.0	0.283
Buxey	29	30	12.0	0.307	12.0	0.462	12.0	0.397	12.0	0.287	12.0	0.235
Buxey	29	33	11.0	0.138	11.0	0.217	11.0	0.292	11.0	0.309	11.0	0.271
Buxey	29	36	10.0	0.204	10.0	0.183	10.0	0.45	10.0	0.36	10.0	0.384
Buxey	29	41	8.0	0.209	8.0	0.316	8.0	0.464	8.0	0.349	8.0	0.481
Buxey	29	47	7.0	0.266	7.0	0.416	7.0	0.491	7.0	0.325	7.0	0.48
Buxey	29	54	7.0	0.156	7.0	0.26	7.0	0.394	7.0	0.175	7.0	0.361
Sawyer	30	25	14.0	1.07	14.0	1.339	14.0	0.435	14.0	0.516	14.0	0.381
Sawyer	30	27	13.0	0.261	13.0	0.326	13.0	0.294	13.0	0.329	13.0	0.234
Sawyer	30	30	12.0	0.367	12.0	0.424	12.0	0.361	12.0	0.474	12.0	0.433
Sawyer	30	33	11.0	0.286	11.0	0.44	11.0	0.559	11.0	0.466	11.0	0.505
Sawyer	30	36	10.0	0.191	10.0	0.379	10.0	0.338	10.0	0.316	10.0	0.298
Sawyer	30	41	8.0	0.252	8.0	0.277	8.0	0.318	8.0	0.301	8.0	0.382
Sawyer	30	47	7.0	0.273	7.0	0.339	7.0	0.439	7.0	0.326	7.0	0.442
Sawyer	30	54	7.0	0.206	7.0	0.279	7.0	0.339	7.0	0.2	7.0	0.246
Sawyer	30	75	5.0	0.116	5.0	0.15	5.0	0.233	5.0	0.194	5.0	0.275
Gunther	35	41	14.0	0.387	14.0	0.395	14.0	1.063	14.0	0.613	14.0	0.857
Gunther	35	44	12.0	0.409	12.0	0.435	12.0	0.809	12.0	0.599	12.0	0.777
Gunther	35	49	11.0	1.096	11.0	1.319	11.0	0.954	11.0	0.71	11.0	0.813
Gunther	35	54	9.0	0.231	9.0	0.334	9.0	0.657	9.0	0.216	9.0	0.506
Gunther	35	61	9.0	0.32	9.0	0.353	9.0	0.516	9.0	0.289	9.0	0.516
Gunther	35	69	8.0	0.209	8.0	0.233	8.0	0.582	8.0	0.352	8.0	0.475
Gunther	35	81	7.0	0.179	7.0	0.233	7.0	0.516	7.0	0.258	7.0	0.507

Kilbridge	45	56	10.0	0.279	10.0	0.518	10.0	1.449	10.0	0.2	10.0	1.136
Kilbridge	45	57	10.0	0.237	10.0	0.471	10.0	1.345	10.0	0.455	10.0	1.07
Kilbridge	45	62	9.0	0.529	9.0	0.677	9.0	2.092	9.0	0.274	9.0	0.91
Kilbridge	45	69	8.0	0.346	8.0	0.337	8.0	1.353	8.0	0.705	8.0	0.859
Kilbridge	45	79	7.0	0.404	7.0	0.408	7.0	1.45	7.0	0.221	7.0	0.76
Kilbridge	45	92	6.0	0.263	6.0	0.433	6.0	0.915	6.0	0.298	6.0	0.817
Kilbridge	45	110	6.0	0.232	6.0	0.217	6.0	0.985	6.0	0.244	6.0	0.639
Kilbridge	45	111	5.0	0.229	5.0	0.217	5.0	0.966	5.0	0.299	5.0	0.671
Kilbridge	45	138	4.0	0.167	4.0	0.233	4.0	0.687	4.0	0.142	4.0	0.846
Kilbridge	45	184	3.0	0.135	3.0	0.166	3.0	0.81	3.0	0.134	3.0	0.584
Hahn	53	2004	8.0	0.283	8.0	0.321	8.0	2.024	8.0	0.298	8.0	1.764
Hahn	53	2338	7.0	0.465	7.0	0.534	7.0	2.15	7.0	0.465	7.0	1.808
Hahn	53	2806	6.0	0.22	6.0	0.265	6.0	1.384	6.0	0.662	6.0	1.561
Hahn	53	3507	5.0	0.25	5.0	0.252	5.0	1.549	5.0	0.244	5.0	1.557
Hahn	53	4676	4.0	0.216	4.0	0.253	4.0	1.354	4.0	0.163	4.0	1.363
Tonge	70	160	23.0	141.688	23.0	175.705	23.0	42.152	23.0	(8.333%)	23.0	43.069
Tonge	70	168	22.0	57.076	22.0	65.63	22.0	49.075	22.0	58.615	22.0	19.485
Tonge	70	176	21.0	59.062	21.0	76.287	21.0	44.092	21.0	(9.091%)	21.0	24.207
Tonge	70	185	20.0	32.918	20.0	24.293	20.0	38.76	20.0	1061.368	20.0	10.302
Tonge	70	195	19.0	24.512	19.0	26.634	19.0	21.239	19.0	7.58	19.0	9.291
Tonge	70	207	18.0	(10.526%)	18.0	(10.526%)	18.0	238.591	18.0	(10.526%)	18.0	812.539
Tonge	70	220	17.0	(11.111%)	17.0	(11.111%)	17.0	168.413	17.0	(11.111%)	17.0	18.552
Tonge	70	234	16.0	21.405	16.0	22.121	16.0	22.529	16.0	153.102	16.0	11.843
Tonge	70	251	14.0	423.714	14.0	335.298	14.0	19.698	15.0	(12.5%)	14.0	39.199
Tonge	70	270	14.0	10.606	14.0	10.919	14.0	19.424	14.0	6.287	14.0	15.635
Tonge	70	293	13.0	1216.308	13.0	(14.286%)	13.0	80.471	13.0	(14.286%)	13.0	260.603
Tonge	70	320	11.0	5.083	11.0	5.827	11.0	7.987	11.0	6.406	11.0	7.037
Tonge	70	364	10.0	1.832	10.0	1.144	10.0	7.448	10.0	0.881	10.0	3.5
Tonge	70	410	9.0	1.781	9.0	0.532	9.0	4.553	9.0	1.235	9.0	3.571
Tonge	70	468	8.0	1.389	8.0	1.044	8.0	6.566	8.0	0.999	8.0	3.213
Tonge	70	527	7.0	0.465	7.0	0.502	7.0	4.698	7.0	0.519	7.0	3.45
Arcus1	83	3786	21.0	6.273	21.0	9.782	21.0	33.289	21.0	6.282	21.0	9.577
Arcus1	83	3985	20.0	7.584	20.0	6.571	20.0	39.11	20.0	4.5	20.0	10.714
Arcus1	83	4206	19.0	4.698	19.0	4.749	19.0	13.988	19.0	3.016	19.0	9.666
Arcus1	83	4454	18.0	2.849	18.0	2.899	18.0	18.199	18.0	2.749	18.0	13.865
Arcus1	83	4732	17.0	6.152	17.0	6.271	17.0	14.026	17.0	1.933	17.0	7.965
Arcus1	83	5048	16.0	7.172	16.0	5.453	16.0	15.473	16.0	1.982	16.0	7.633
Arcus1	83	5408	15.0	4.509	15.0	4.51	15.0	13.132	15.0	1.951	15.0	8.232
Arcus1	83	5824	14.0	4.422	14.0	2.514	14.0	10.648	14.0	2.416	14.0	7.632
Arcus1	83	5853	14.0	2.383	14.0	2.332	14.0	10.868	14.0	2.642	14.0	7.266
Arcus1	83	6309	13.0	4.48	13.0	4.472	13.0	21.515	13.0	2.567	13.0	6.964
Arcus1	83	6842	12.0	2.332	12.0	2.448	12.0	10.378	12.0	1.995	12.0	6.748
Arcus1	83	6883	12.0	2.749	12.0	2.95	12.0	10.371	12.0	1.331	12.0	6.915
Arcus1	83	7571	11.0	2.138	11.0	2.015	11.0	10.876	11.0	1.546	11.0	6.59
Arcus1	83	8412	10.0	1.999	10.0	2.03	10.0	9.548	10.0	1.039	10.0	6.224
Arcus1	83	8898	9.0	1.733	9.0	2.048	9.0	9.149	9.0	1.232	9.0	6.081
Arcus1	83	10816	8.0	2.373	8.0	2.432	8.0	9.075	8.0	1.732	8.0	9.735
Arcus2	111	5755	27.0	15.514	27.0	15.6	27.0	181.878	27.0	6.399	27.0	34.58
Arcus2	111	5785	27.0	6.794	27.0	6.865	27.0	113.438	27.0	6.307	27.0	95.658
Arcus2	111	6016	26.0	14.159	26.0	13.972	26.0	169.102	26.0	6.795	26.0	22.033
Arcus2	111	6267	25.0	8.678	25.0	8.469	25.0	157.28	25.0	3.298	25.0	24.079
Arcus2	111	6540	24.0	6.956	24.0	7.217	24.0	105.495	24.0	3.612	24.0	71.197
Arcus2	111	6837	23.0	6.614	23.0	6.448	23.0	94.397	23.0	3.742	23.0	87.732
Arcus2	111	7162	22.0	6.349	22.0	6.365	22.0	86.02	22.0	5.415	22.0	77.229
Arcus2	111	7520	21.0	(9.091%)	21.0	(9.091%)	21.0	(9.091%)	21.0	(9.091%)	21.0	(9.091%)
Arcus2	111	7916	20.0	5.657	20.0	5.732	20.0	89.533	20.0	3.268	20.0	44.1
Arcus2	111	8356	19.0	5.964	19.0	6.216	19.0	70.842	19.0	4.272	19.0	29.068
Arcus2	111	8847	18.0	5.701	18.0	5.695	18.0	150.677	18.0	2.976	18.0	108.166
Arcus2	111	9400	17.0	5.033	17.0	5.25	17.0	52.263	17.0	2.482	17.0	31.049

Arcus2	111	10027	16.0	5.332	16.0	5.488	16.0	23.886	16.0	2.516	16.0	37.733
Arcus2	111	10743	15.0	7.375	15.0	7.55	15.0	44.832	15.0	2.516	15.0	36.024
Arcus2	111	11378	14.0	4.832	14.0	5.178	14.0	24.281	14.0	3.131	14.0	13.908
Arcus2	111	11570	14.0	(13.333%)	14.0	(13.333%)	14.0	(13.333%)	14.0	(13.333%)	14.0	(13.333%)
Arcus2	111	17067	9.0	2.234	9.0	2.516	9.0	29.515	9.0	2.108	9.0	13.133

Table A.1: Complete performance results for SALBP-1.

A.2 TALBP-1

Problem	n	c	MILP-1		MILP-2		CP	
			Obj.	CPU (s)	Obj.	CPU (s)	Obj.	CPU (s)
P9	9	5	2[4]	0.526	2[4]	0.106	2[4]	0.084
P9	9	6	2[3]	0.425	2[3]	0.08	2[3]	0.021
P12	12	5	3[6]	13.465	3[6]	0.163	3[6]	0.036
P12	12	6	3[5]	25.858	3[5]	0.121	3[5]	0.07
P12	12	7	2[4]	1.262	2[4]	0.134	2[4]	0.019
P12	12	8	2[4]	1.681	2[4]	0.132	2[4]	0.015
P16	16	16	3[6]	11.955	3[6]	0.238	3[6]	0.023
P16	16	19	3[5]	19.777	3[5]	0.192	3[5]	0.076
P16	16	22	2[4]	2.296	2[4]	0.207	2[4]	0.021
P24	24	20	4[7]	(25.123%)	4[7]	1.273	4[7]	0.089
P24	24	25	3[6]	(24.183%)	3[6]	1.307	3[6]	0.057
P24	24	30	3[5]	(32.895%)	3[5]	0.589	3[5]	0.096
P24	24	35	2[4]	546.84	2[4]	0.849	2[4]	0.056
P24	24	40	2[4]	(0.98%)	2[4]	0.515	2[4]	0.053
P65	65	326	9[17]	(77.759%)	9[16]	(10.962%)	8[16]	26.028
P65	65	381	8[14]	(75.047%)	7[14]	435.29	7[14]	4.552
P65	65	435	7[13]	(71.505%)	7[12]	432.499	6[12]	7.377
P65	65	490	6[11]	(83.312%)	6[11]	168.732	6[11]	1.059
P65	65	544	6[10]	(83.291%)	5[10]	162.976	5[10]	1.044
P148	148	255	-	-	-	-	11[21]	12.279
P148	148	306	-	-	-	-	9[17]	12.296
P148	148	357	-	-	-	-	8[15]	12.014
P148	148	408	-	-	-	-	7[13]	9.188
P148	148	459	-	-	-	-	6[12]	7.774
P148	148	510	-	-	-	-	6[11]	9.224
P205	205	1133	-	-	-	-	11[21]	293.97
P205	205	1322	-	-	-	-	9[18]	588.798
P205	205	1510	-	-	-	-	8[16]	42.753
P205	205	1699	-	-	-	-	7[14]	17.737
P205	205	1888	-	-	-	-	7[13]	26.388
P205	205	2077	-	-	-	-	6[12]	23.794

P205	205	2266	-	-	-	-	6[11]	48.685
P205	205	2454	-	-	-	-	5[10]	19.292
P205	205	2643	-	-	-	-	5[9]	58.16
P205	205	2832	-	-	-	-	5[9]	21.213

Table A.2: Benchmark results for TALBP-1. Objective values are denoted as the number of mated-stations [number of individual stations]. e.g. 2[4] means two mated-stations and 4 individual stations.

A.3 HRALBP-1

Problem	n	c	n^h	n^r	MILP		CP	
					Obj.	CPU (s)	Obj.	CPU (s)
Bowman	8	25	0	1	4.0	0.095	4.0	0.104
Bowman	8	25	0	2	4.0	0.066	4.0	0.032
Bowman	8	25	1	0	2.0	0.029	2.0	0.049
Bowman	8	25	1	1	2.0	0.086	2.0	0.124
Bowman	8	25	1	2	2.0	0.136	2.0	0.113
Bowman	8	25	2	0	2.0	0.076	2.0	0.049
Bowman	8	25	2	1	2.0	0.051	2.0	0.11
Bowman	8	25	2	2	2.0	0.064	2.0	0.116
Jackson	11	8	0	1	7.0	0.131	7.0	0.07
Jackson	11	8	0	2	5.0	0.088	5.0	0.056
Jackson	11	8	1	0	3.0	0.174	3.0	0.087
Jackson	11	8	1	1	3.0	0.299	3.0	0.216
Jackson	11	8	1	2	2.0	0.274	2.0	0.208
Jackson	11	8	2	0	3.0	0.224	3.0	0.099
Jackson	11	8	2	1	2.0	0.363	2.0	0.235
Jackson	11	8	2	2	2.0	0.305	2.0	0.195
Roszieg	25	22	0	1	6.0	3.905	6.0	0.909
Roszieg	25	22	0	2	5.0	6.837	5.0	1.205
Roszieg	25	22	1	0	3.0	2.099	3.0	0.912
Roszieg	25	22	1	1	3.0	9.543	3.0	4.89
Roszieg	25	22	1	2	2.0	4.884	2.0	4.624
Roszieg	25	22	2	0	3.0	16.661	3.0	1.479
Roszieg	25	22	2	1	2.0	3.089	2.0	4.273
Roszieg	25	22	2	2	2.0	3.321	2.0	5.191
Case study	28	43	0	1	6.0	(50.0%)	6.0	2.456
Case study	28	43	0	2	4.0	(25.0%)	4.0	34.714
Case study	28	43	1	0	3.0	22.293	3.0	1.769

Case study	28	43	1	1	2.0	23.948	2.0	6.924
Case study	28	43	1	2	2.0	19.716	2.0	7.934
Case study	28	43	2	0	2.0	5.732	2.0	1.371
Case study	28	43	2	1	2.0	5.194	2.0	7.439
Case study	28	43	2	2	2.0	149.272	2.0	14.002
Buxey	29	55	0	1	6.0	29.548	6.0	2.687
Buxey	29	55	0	2	5.0	15.448	5.0	2.416
Buxey	29	55	1	0	3.0	3.639	3.0	1.74
Buxey	29	55	1	1	3.0	(33.333%)	3.0	15.88
Buxey	29	55	1	2	2.0	15.143	2.0	9.271
Buxey	29	55	2	0	3.0	(33.333%)	3.0	4.408
Buxey	29	55	2	1	2.0	12.554	2.0	8.683
Buxey	29	55	2	2	2.0	5.184	2.0	9.094
Sawyer	30	38	0	1	9.0	310.021	9.0	2.854
Sawyer	30	38	0	2	7.0	12.555	7.0	1.886
Sawyer	30	38	1	0	5.0	(20.0%)	5.0	3.138
Sawyer	30	38	1	1	3.0	32.02	3.0	11.167
Sawyer	30	38	1	2	3.0	16.759	3.0	9.564
Sawyer	30	38	2	0	3.0	11.969	3.0	2.398
Sawyer	30	38	2	1	3.0	15.207	3.0	11.547
Sawyer	30	38	2	2	3.0	24.991	3.0	10.432
Gunther	35	81	0	1	7.0	1553.126	7.0	5.399
Gunther	35	81	0	2	6.0	582.915	6.0	5.467
Gunther	35	81	1	0	3.0	28.376	3.0	3.179
Gunther	35	81	1	1	3.0	42.445	3.0	32.422
Gunther	35	81	1	2	3.0	140.818	3.0	32.367
Gunther	35	81	2	0	3.0	52.875	3.0	5.793
Gunther	35	81	2	1	2.0	12.813	2.0	21.691
Gunther	35	81	2	2	2.0	13.893	2.0	15.743
Kilbridge	45	92	0	1	6.0	(33.333%)	6.0	14.642
Kilbridge	45	92	0	2	5.0	(40.0%)	5.0	1886.598
Kilbridge	45	92	1	0	3.0	(33.333%)	3.0	8.812
Kilbridge	45	92	1	1	3.0	(33.333%)	3.0	(33.333%)
Kilbridge	45	92	1	2	2.0	70.396	2.0	47.074
Kilbridge	45	92	2	0	3.0	(33.333%)	3.0	(33.333%)
Kilbridge	45	92	2	1	2.0	129.123	2.0	34.261
Kilbridge	45	92	2	2	2.0	63.052	2.0	22.258
Hahn	53	2806	0	1	6.0	34.027	6.0	19.549
Hahn	53	2806	0	2	5.0	132.383	5.0	20.707
Hahn	53	2806	1	0	3.0	26.737	3.0	14.32
Hahn	53	2806	1	1	3.0	(33.333%)	3.0	84.067

Hahn	53	2806	1	2	3.0	41.492	2.0	81.311
Hahn	53	2806	2	0	3.0	24.363	3.0	34.49
Hahn	53	2806	2	1	3.0	40.378	2.0	74.458
Hahn	53	2806	2	2	2.0	27.559	2.0	52.347
Tonge	70	1170	0	1	-	-	3.0	41.564
Tonge	70	1170	0	2	3.0	340.049	2.0	98.366
Tonge	70	1170	1	0	2.0	(50.0%)	2.0	41.762
Tonge	70	1170	1	1	2.0	(50.0%)	1.0	123.057
Tonge	70	1170	1	2	1.0	314.175	1.0	147.523
Tonge	70	1170	2	0	1.0	918.71	1.0	48.033
Tonge	70	1170	2	1	1.0	136.308	1.0	183.908
Tonge	70	1170	2	2	1.0	181.762	1.0	115.993
Arcus1	83	25236	0	1	-	-	3.0	76.951
Arcus1	83	25236	0	2	3.0	1301.599	3.0	(33.333%)
Arcus1	83	25236	1	0	3.0	257.656	2.0	61.446
Arcus1	83	25236	1	1	-	-	2.0	707.574
Arcus1	83	25236	1	2	2.0	1028.664	1.0	155.165
Arcus1	83	25236	2	0	2.0	202.3	2.0	380.091
Arcus1	83	25236	2	1	1.0	2002.176	1.0	131.767
Arcus1	83	25236	2	2	2.0	1208.541	1.0	168.665
Arcus2	111	50133	0	1	-	-	3.0	186.317
Arcus2	111	50133	0	2	-	-	3.0	(33.333%)
Arcus2	111	50133	1	0	3.0	2294.634	2.0	142.571
Arcus2	111	50133	1	1	-	-	2.0	1892.278
Arcus2	111	50133	1	2	-	-	1.0	201.154
Arcus2	111	50133	2	0	111.0	(97.297%)	2.0	381.068
Arcus2	111	50133	2	1	17.0	(88.235%)	1.0	238.712
Arcus2	111	50133	2	2	31.0	(93.548%)	1.0	383.53

Table A.3: Complete performance results for HRALBP-1.

A.4 HRALBP-2

Problem	n	m	n^h	n^r	MILP		CP	
					Obj.	CPU (s)	Obj.	CPU (s)
Bowman	8	2	0	1	38.0	0.092	38.0	0.123
Bowman	8	2	0	2	35.0	0.038	35.0	0.043
Bowman	8	2	1	0	19.0	0.036	19.0	0.042
Bowman	8	2	1	1	17.5	0.065	17.5	0.074
Bowman	8	2	1	2	17.5	0.051	17.5	0.074
Bowman	8	2	2	0	17.5	0.04	17.5	0.02

Bowman	8	2	2	1	17.5	0.062	17.5	0.051
Bowman	8	2	2	2	17.5	0.065	17.5	0.051
Jackson	11	2	0	1	23.0	0.101	23.0	0.042
Jackson	11	2	0	2	16.0	0.158	16.0	0.038
Jackson	11	2	1	0	11.5	0.152	11.5	0.027
Jackson	11	2	1	1	8.5	0.133	8.5	0.087
Jackson	11	2	1	2	8.0	0.184	8.0	0.068
Jackson	11	2	2	0	8.5	0.188	8.5	0.041
Jackson	11	2	2	1	8.0	0.185	8.0	0.058
Jackson	11	2	2	2	8.0	0.158	8.0	0.078
Roszieg	25	3	0	1	42.0	65.627	42.0	0.133
Roszieg	25	3	0	2	32.0	3.435	32.0	0.227
Roszieg	25	3	1	0	21.0	32.491	21.0	0.114
Roszieg	25	3	1	1	16.0	7.824	16.0	0.631
Roszieg	25	3	1	2	15.0	4.899	15.0	0.421
Roszieg	25	3	2	0	16.0	3.704	16.0	0.235
Roszieg	25	3	2	1	15.0	4.761	15.0	0.816
Roszieg	25	3	2	2	15.0	4.145	15.0	0.59
Case study	28	2	0	1	108.5	(45.53%)	108.5	0.112
Case study	28	2	0	2	65.6	(32.927%)	65.5	383.376
Case study	28	2	1	0	57.4	(28.672%)	57.4	0.17
Case study	28	2	1	1	34.0	(17.647%)	34.0	2.767
Case study	28	2	1	2	26.6	(5.639%)	26.6	1.438
Case study	28	2	2	0	34.0	(17.647%)	34.0	7.519
Case study	28	2	2	1	24.6	14.496	24.6	0.628
Case study	28	2	2	2	24.6	10.025	24.6	0.373
Buxey	29	3	0	1	108.0	(25.926%)	108.0	0.248
Buxey	29	3	0	2	75.0	202.376	75.0	0.607
Buxey	29	3	1	0	54.0	(24.074%)	54.0	0.285
Buxey	29	3	1	1	37.5	2476.903	37.5	1.316
Buxey	29	3	1	2	34.0	13.72	34.0	0.87
Buxey	29	3	2	0	37.5	302.713	37.5	0.575
Buxey	29	3	2	1	32.5	5.221	32.5	0.692
Buxey	29	3	2	2	32.5	7.244	32.5	0.769
Sawyer	30	4	0	1	81.0	(35.802%)	81.0	0.589
Sawyer	30	4	0	2	55.0	(9.091%)	55.0	1.431
Sawyer	30	4	1	0	40.5	(40.733%)	40.5	0.646
Sawyer	30	4	1	1	27.5	(9.091%)	27.5	3.235
Sawyer	30	4	1	2	25.0	45.505	25.0	0.935
Sawyer	30	4	2	0	27.5	(9.091%)	27.5	2.058
Sawyer	30	4	2	1	23.5	15.432	23.5	0.767

Sawyer	30	4	2	2	23.5	13.799	23.5	0.744
Gunther	35	4	0	1	121.0	(14.876%)	121.0	0.533
Gunther	35	4	0	2	89.0	52.866	89.0	0.673
Gunther	35	4	1	0	60.5	(30.506%)	60.5	0.382
Gunther	35	4	1	1	44.5	118.936	44.5	1.221
Gunther	35	4	1	2	44.0	55.874	44.0	1.698
Gunther	35	4	2	0	44.5	115.112	44.5	0.854
Gunther	35	4	2	1	43.5	37.532	43.5	1.411
Gunther	35	4	2	2	43.5	30.194	43.5	1.448
Kilbridge	45	5	0	1	111.0	(50.45%)	111.0	5.005
Kilbridge	45	5	0	2	76.0	(27.632%)	76.0	1118.114
Kilbridge	45	5	1	0	55.5	(50.45%)	55.5	4.238
Kilbridge	45	5	1	1	39.0	(29.487%)	38.0	1475.105
Kilbridge	45	5	1	2	34.0	(19.118%)	34.0	13.214
Kilbridge	45	5	2	0	38.5	(28.571%)	38.0	928.051
Kilbridge	45	5	2	1	30.0	(8.333%)	30.0	9.559
Kilbridge	45	5	2	2	29.0	(5.172%)	29.0	3.306
Hahn	53	6	0	1	2400.0	(0.958%)	2400.0	1.19
Hahn	53	6	0	2	2242.0	566.843	2242.0	1.952
Hahn	53	6	1	0	1200.0	(0.208%)	1200.0	1.276
Hahn	53	6	1	1	1121.0	134.372	1121.0	4.166
Hahn	53	6	1	2	1096.5	35.235	1096.5	4.05
Hahn	53	6	2	0	1121.0	92.454	1121.0	2.854
Hahn	53	6	2	1	1096.5	47.535	1096.5	4.619
Hahn	53	6	2	2	1096.5	39.523	1096.5	4.462
Tonge	70	3	0	1	1171.0	(52.178%)	1170.0	0.843
Tonge	70	3	0	2	790.0	(37.595%)	781.0	(2.074%)
Tonge	70	3	1	0	585.0	(49.06%)	585.0	2.091
Tonge	70	3	1	1	398.0	(37.94%)	390.5	(1.997%)
Tonge	70	3	1	2	345.5	(28.329%)	344.5	(5.196%)
Tonge	70	3	2	0	396.0	(34.227%)	390.5	(1.767%)
Tonge	70	3	2	1	318.5	(22.603%)	316.5	(6.951%)
Tonge	70	3	2	2	304.5	(10.509%)	304.5	(5.353%)
Arcus1	83	3	0	1	25237.0	(41.443%)	25236.0	32.253
Arcus1	83	3	0	2	18062.0	(23.857%)	17895.0	(12.732%)
Arcus1	83	3	1	0	12619.5	(43.738%)	12618.0	20.592
Arcus1	83	3	1	1	8998.5	(43.013%)	8947.0	(15.747%)
Arcus1	83	3	1	2	8043.0	(39.339%)	7964.5	(8.534%)
Arcus1	83	3	2	0	8995.0	(27.343%)	8949.5	(11.458%)
Arcus1	83	3	2	1	7573.5	(32.29%)	7219.0	(2.309%)
Arcus1	83	3	2	2	7129.0	(7.596%)	7044.5	23.557

Arcus2	111	3	0	1	50145.0	(53.609%)	50133.0	14.288
Arcus2	111	3	0	2	35680.0	(35.819%)	34113.0	(7.563%)
Arcus2	111	3	1	0	25078.0	(49.952%)	25066.5	10.569
Arcus2	111	3	1	1	19235.0	(52.507%)	17056.5	(15.52%)
Arcus2	111	3	1	2	15355.5	(31.139%)	14768.5	(12.51%)
Arcus2	111	3	2	0	17464.0	(33.566%)	17056.5	(7.58%)
Arcus2	111	3	2	1	13893.0	(25.887%)	13052.0	(1.008%)
Arcus2	111	3	2	2	12920.5	(19.132%)	12655.0	49.307

Table A.4: Complete performance results for HRALBP-2.

A.5 HRALBP-3

Problem	n	m	n^h	n^r	MILP		CP	
					Obj.	CPU (s)	Obj.	CPU (s)
Bowman	8	25	1	1	4351.4	0.237	4351.4	0.42
Bowman	8	25	1	2	4351.4	0.169	4351.4	0.332
Bowman	8	25	2	1	4351.4	0.216	4351.4	0.413
Bowman	8	25	2	2	4351.4	0.279	4351.4	0.409
Jackson	11	8	1	1	6202.2	0.587	6202.2	0.546
Jackson	11	8	1	2	6202.2	0.466	6202.2	0.545
Jackson	11	8	2	1	6202.2	0.95	6202.2	0.533
Jackson	11	8	2	2	5551.7	0.649	5551.7	0.584
Roszieg	25	22	1	1	6354.7	352.035	6354.7	19.497
Roszieg	25	22	1	2	5360.9	79.479	5360.9	20.126
Roszieg	25	22	2	1	5204.8	1762.337	5204.8	23.104
Roszieg	25	22	2	2	5204.8	97.069	5204.8	23.152
Case study	28	43	1	1	4509.9	(3.396%)	4509.9	21.364
Case study	28	43	1	2	4509.9	(0.024%)	4509.9	20.165
Case study	28	43	2	1	4509.9	(3.397%)	4509.9	20.137
Case study	28	43	2	2	4509.9	(28.93%)	4509.9	51.068
Buxey	29	55	1	1	6351.9	(21.179%)	6351.9	39.994
Buxey	29	55	1	2	5379.3	(16.284%)	5379.3	52.624
Buxey	29	55	2	1	5051.9	(6.903%)	4864.6	43.772
Buxey	29	55	2	2	4864.6	(10.452%)	4864.6	47.843
Sawyer	30	38	1	1	7526.4	(13.579%)	7526.4	64.788
Sawyer	30	38	1	2	7526.4	(15.572%)	7526.4	71.207
Sawyer	30	38	2	1	7205.8	(11.815%)	7205.5	72.683
Sawyer	30	38	2	2	7207.2	(11.832%)	7205.5	73.233
Gunther	35	81	1	1	6350.6	765.85	6350.6	56.633
Gunther	35	81	1	2	6350.6	(2.362%)	6350.6	64.952

Gunther	35	81	2	1	5212.7	(0.232%)	5212.7	69.996
Gunther	35	81	2	2	5212.7	(6.799%)	5212.7	62.788
Kilbridge	45	92	1	1	6350.9	(33.853%)	6350.9	3321.017
Kilbridge	45	92	1	2	6350.9	(33.853%)	5401.1	(6.489%)
Kilbridge	45	92	2	1	5054.1	(16.88%)	4875.3	(0.079%)
Kilbridge	45	92	2	2	5053.8	(16.875%)	4875.3	(0.079%)
Hahn	53	2806	1	1	6200.2	1767.702	6200.2	127.443
Hahn	53	2806	1	2	6200.2	(21.758%)	6200.2	170.373
Hahn	53	2806	2	1	5050.5	(6.936%)	5050.5	165.286
Hahn	53	2806	2	2	5050.5	(6.936%)	5050.5	277.154
Tonge	70	1170	1	1	-	-	4200.4	242.32
Tonge	70	1170	1	2	-	-	4200.4	(32.07%)
Tonge	70	1170	2	1	2700.4	(12.961%)	2700.4	140.625
Tonge	70	1170	2	2	2700.4	(12.961%)	2700.4	171.401
Arcus1	83	25236	1	1	-	-	4200.1	437.767
Arcus1	83	25236	1	2	-	-	4200.1	(31.976%)
Arcus1	83	25236	2	1	-	-	2701.5	(0.01%)
Arcus1	83	25236	2	2	4200.4	(44.049%)	2701.5	(0.01%)
Arcus2	111	50133	1	1	-	-	4200.1	1509.232
Arcus2	111	50133	1	2	-	-	4200.1	(31.817%)
Arcus2	111	50133	2	1	-	-	2701.0	(0.03%)
Arcus2	111	50133	2	2	-	-	2701.0	(6.926%)

Table A.5: Complete performance results for HRALBP-3.

Appendix B

Solution Logs

B.1 TALBP-1 MILP-1 Log

Gurobi log for P65 instance with $c = 490$.

```
Gurobi 12.0.3 (win64) logging started Mon Oct 13 15:07:16 2025

Set parameter OutputFlag to value 1
Set parameter TimeLimit to value 3600
Gurobi Optimizer version 12.0.3 build v12.0.3rc0 (win64 - Windows 11.0 (26100.2))

CPU model: 13th Gen Intel(R) Core(TM) i7-13700H, instruction set [SSE2|AVX|AVX2]
Thread count: 14 physical cores, 20 logical processors, using up to 20 threads

Non-default parameters:
TimeLimit  3600
MIPGap  0
LogToConsole  0

Optimize a model with 155962 rows, 9686 columns and 813800 nonzeros
Model fingerprint: 0xac864962
Variable types: 65 continuous, 9621 integer (9621 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+03]
  Objective range  [8e-03, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 3e+03]
Presolve removed 71 rows and 1827 columns
Presolve time: 0.92s
Presolved: 155891 rows, 7859 columns, 799246 nonzeros
Variable types: 64 continuous, 7795 integer (7736 binary)
Root relaxation presolved: 7794 rows, 163685 columns, 807105 nonzeros

Root relaxation: objective 6.497710e-01, 3870 iterations, 1.02 seconds (1.81 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.64977    0  106          -    0.64977      -     -    3s
     0     0    0.64977    0  239          -    0.64977      -     -    5s
     0     0    0.64977    0  211          -    0.64977      -     -    6s
     0     0    0.64977    0  145          -    0.64977      -     -   13s
```

	0	0	0.64977	0	245	-	0.64977	-	-	14s
	0	0	0.65339	0	120	-	0.65339	-	-	30s
	0	0	0.65339	0	283	-	0.65339	-	-	30s
	0	0	0.65339	0	363	-	0.65339	-	-	32s
	0	0	0.65339	0	359	-	0.65339	-	-	33s
	0	0	0.65339	0	126	-	0.65339	-	-	61s
	0	0	0.65339	0	329	-	0.65339	-	-	62s
	0	0	0.65339	0	143	-	0.65339	-	-	85s
	0	0	0.65339	0	314	-	0.65339	-	-	86s
	0	0	0.65339	0	135	-	0.65339	-	-	96s
	0	0	0.65339	0	307	-	0.65339	-	-	98s
H	0	0				15.1450382	0.65339	95.7%	-	110s
H	0	0				14.1374046	0.65339	95.4%	-	110s
	0	0	0.65339	0	142	14.13740	0.65339	95.4%	-	110s
	0	0	0.65339	0	142	14.13740	0.65339	95.4%	-	111s
H	0	0				13.1374046	1.01527	92.3%	-	116s
	0	2	1.01527	0	118	13.13740	1.01527	92.3%	-	119s
	1	4	1.01527	1	201	13.13740	1.01527	92.3%	2768	122s
	3	8	1.01527	2	275	13.13740	1.01527	92.3%	2413	125s
	7	16	1.01527	3	262	13.13740	1.01527	92.3%	1977	131s
	15	30	1.01527	4	277	13.13740	1.01527	92.3%	1814	137s
H	29	74				13.1297710	1.01527	92.3%	1559	144s
H	34	74				11.1297710	1.01527	90.9%	1422	144s
H	39	74				11.1221374	1.01527	90.9%	1326	144s
H	65	74				9.1221374	1.01527	88.9%	924	144s
	73	126	1.01527	10	274	9.12214	1.01527	88.9%	856	147s
	125	169	1.01527	13	274	9.12214	1.01527	88.9%	559	150s
H	129	169				8.1068702	1.01527	87.5%	563	150s
	212	228	1.01527	18	263	8.10687	1.01527	87.5%	376	156s
H	215	228				7.0992366	1.01527	85.7%	377	156s
	274	292	1.01527	22	246	7.09924	1.01527	85.7%	309	160s
H	282	292				6.0916031	1.01527	83.3%	305	160s
	359	378	1.01527	29	234	6.09160	1.01527	83.3%	255	165s
	403	431	1.01527	33	251	6.09160	1.01527	83.3%	243	170s
	430	445	1.01527	35	259	6.09160	1.01527	83.3%	240	175s
	461	484	1.01527	38	235	6.09160	1.01527	83.3%	244	180s
	506	538	1.01527	41	220	6.09160	1.01527	83.3%	244	185s
	583	636	1.01527	45	213	6.09160	1.01527	83.3%	224	190s
	707	756	1.01527	49	198	6.09160	1.01527	83.3%	201	196s
	831	899	1.01527	52	191	6.09160	1.01527	83.3%	186	202s
	907	964	1.01527	53	185	6.09160	1.01527	83.3%	182	207s
	974	1028	1.01527	54	180	6.09160	1.01527	83.3%	181	210s
	1154	1228	1.01527	57	219	6.09160	1.01527	83.3%	173	219s
	1265	1314	1.01527	59	184	6.09160	1.01527	83.3%	170	224s
	1353	1414	1.01527	61	209	6.09160	1.01527	83.3%	169	227s
	1458	1533	1.01527	62	209	6.09160	1.01527	83.3%	162	231s
	1579	1688	1.01527	64	172	6.09160	1.01527	83.3%	158	238s
	1741	1834	1.02765	64	225	6.09160	1.01527	83.3%	150	242s
	1904	1992	1.02766	65	214	6.09160	1.01527	83.3%	142	247s
	2086	2063	1.02825	76	220	6.09160	1.01527	83.3%	136	254s
	2198	2162	1.02850	77	222	6.09160	1.01527	83.3%	136	259s
	2321	2245	1.03051	85	284	6.09160	1.01527	83.3%	134	266s
	2412	2350	1.03071	92	286	6.09160	1.01527	83.3%	138	272s
	2526	2476	1.03828	101	258	6.09160	1.01527	83.3%	137	279s
	2664	2607	1.03923	105	187	6.09160	1.01527	83.3%	136	284s
	2812	2691	1.03955	114	243	6.09160	1.01527	83.3%	134	293s
	2911	2778	2.02478	125	207	6.09160	1.01527	83.3%	137	300s
	3004	2871	1.04161	131	249	6.09160	1.01527	83.3%	139	307s
	3101	2957	1.04294	136	259	6.09160	1.01527	83.3%	139	315s

	3206	3029	1.04412	143	252	6.09160	1.01527	83.3%	139	324s
	3292	3198	1.04494	147	255	6.09160	1.01527	83.3%	138	330s
	3462	3378	1.04533	153	242	6.09160	1.01527	83.3%	135	339s
	3655	3546	1.04809	159	254	6.09160	1.01527	83.3%	133	347s
	3834	3653	1.04981	168	244	6.09160	1.01527	83.3%	132	356s
	3967	3788	2.02939	178	201	6.09160	1.01527	83.3%	136	367s
	4143	3789	3.03053	81	142	6.09160	1.01527	83.3%	139	395s
	4145	3790	5.05344	222	265	6.09160	1.01527	83.3%	139	448s
	4146	3791	2.03053	76	512	6.09160	1.01527	83.3%	139	464s
	4147	3792	5.05344	239	701	6.09160	1.01527	83.3%	139	479s
	4148	3792	1.03042	53	190	6.09160	1.01527	83.3%	139	542s
	4149	3793	1.13777	162	746	6.09160	1.01527	83.3%	139	567s
	4150	3794	5.05344	224	206	6.09160	1.01527	83.3%	139	611s
	4151	3794	1.14977	39	206	6.09160	1.01527	83.3%	139	622s
	4152	3795	2.23595	41	206	6.09160	1.01527	83.3%	139	646s
	4153	3799	1.01527	11	564	6.09160	1.01527	83.3%	183	687s
	4155	3802	1.01527	12	549	6.09160	1.01527	83.3%	185	706s
	4159	3809	1.01527	13	689	6.09160	1.01527	83.3%	193	728s
	4167	3819	1.01527	14	594	6.09160	1.01527	83.3%	205	762s
	4180	3827	1.01527	15	691	6.09160	1.01527	83.3%	226	790s
	4192	3857	1.01527	16	576	6.09160	1.01527	83.3%	230	840s
	4227	3859	1.05070	17	429	6.09160	1.01527	83.3%	252	874s
	4241	3870	1.01527	18	541	6.09160	1.01527	83.3%	265	912s
	4259	3885	1.05177	18	584	6.09160	1.01527	83.3%	289	949s
	4280	3907	1.05814	19	393	6.09160	1.01527	83.3%	310	980s
	4315	3920	2.05402	21	314	6.09160	1.01527	83.3%	330	1015s
	4350	3939	1.05821	20	453	6.09160	1.01527	83.3%	344	1046s
	4393	3949	1.53242	21	375	6.09160	1.01527	83.3%	365	1069s
	4417	3960	infeasible	21		6.09160	1.01527	83.3%	370	1093s
	4443	3976	1.56513	22	330	6.09160	1.01527	83.3%	377	1117s
H	4449	3784				6.0839695	1.01527	83.3%	379	1117s
	4479	3814	1.59204	23	267	6.08397	1.01527	83.3%	381	1136s
	4526	3842	2.09548	25	247	6.08397	1.01527	83.3%	385	1154s
	4585	3875	3.05160	27	255	6.08397	1.01527	83.3%	392	1171s
	4655	3924	3.07926	30	179	6.08397	1.01527	83.3%	394	1188s
	4744	3938	3.07926	34	159	6.08397	1.01527	83.3%	394	1208s
	4795	3999	3.54580	36	120	6.08397	1.01527	83.3%	394	1226s
	4899	4068	4.05344	40	116	6.08397	1.01527	83.3%	392	1243s
	5007	4105	4.04580	44	118	6.08397	1.01527	83.3%	391	1264s
	5116	4103	4.04580	48	116	6.08397	1.01527	83.3%	389	1286s
	5171	4186	4.04580	50	114	6.08397	1.01527	83.3%	388	1305s
	5298	4245	4.04580	56	101	6.08397	1.01527	83.3%	384	1329s
	5436	4233	4.04580	61	90	6.08397	1.01527	83.3%	381	1345s
	5482	4256	4.04580	66	91	6.08397	1.01527	83.3%	383	1368s
	5545	4348	4.04580	68	89	6.08397	1.01527	83.3%	381	1391s
	5716	4397	4.04580	78	82	6.08397	1.01527	83.3%	378	1413s
	5853	4454	4.05543	84	112	6.08397	1.01527	83.3%	377	1439s
	5985	4526	4.55344	88	80	6.08397	1.01527	83.3%	378	1468s
	6127	4635	5.06107	101	61	6.08397	1.01527	83.3%	377	1494s
	6283	4583	1.01527	20	431	6.08397	1.01527	83.3%	374	1495s
	6314	4730	infeasible	115		6.08397	1.01527	83.3%	375	1519s
	6528	4670	5.56870	128	68	6.08397	1.01527	83.3%	371	1545s
	6542	4678	5.56870	129	59	6.08397	1.01527	83.3%	371	1584s
	6561	4686	infeasible	129		6.08397	1.01527	83.3%	373	1606s
	6579	4691	infeasible	130		6.08397	1.01527	83.3%	374	1637s
	6601	4736	2.04596	19	231	6.08397	1.01527	83.3%	376	1663s
	6654	4782	5.05344	107	78	6.08397	1.01527	83.3%	375	1690s
	6745	4806	2.04711	20	247	6.08397	1.01527	83.3%	374	1724s
	6831	4940	2.07734	21	242	6.08397	1.01527	83.3%	380	1752s

```
  7015  4984    3.04580   22  181    6.08397    1.01527  83.3%   377 1786s
  7187  5060    3.04580   23  300    6.08397    1.01527  83.3%   377 1817s
  7371  5137    3.04580   27  206    6.08397    1.01527  83.3%   377 1853s
  7592  5146    3.06167   33  205    6.08397    1.01527  83.3%   378 1895s
  7720  5314    3.06263   39  184    6.08397    1.01527  83.3%   386 1934s
  7956  5404    3.57400   46  184    6.08397    1.01527  83.3%   386 1975s
  8185  5521    5.06870   63  102    6.08397    1.01527  83.3%   390 2018s
  8408  5682    1.56377   25  315    6.08397    1.01527  83.3%   394 2059s
  8689  5863    1.08586   27  347    6.08397    1.01527  83.3%   396 2104s
  8998  5952    1.58991   30  338    6.08397    1.01527  83.3%   396 2154s
  9275  6024    3.54679   34  185    6.08397    1.01527  83.3%   401 2205s
  9511  6106    1.53969   27  449    6.08397    1.01527  83.3%   410 2255s
  9704  6367    1.53938   28  250    6.08397    1.01527  83.3%   417 2307s
 10077  6637    2.53053   30  225    6.08397    1.01527  83.3%   415 2358s
 10576  6887    3.04580   41  120    6.08397    1.01527  83.3%   412 2416s
 11101  7058    4.05840  106   85    6.08397    1.01527  83.3%   409 2473s
 11569  7218    5.12182  173   59    6.08397    1.01527  83.3%   406 2535s
 11987  7488    2.04258   29  222    6.08397    1.01527  83.3%   407 2605s
 12537  7590    4.05133   53  119    6.08397    1.01527  83.3%   404 2679s
 12962  7841    2.03532   29  252    6.08397    1.01527  83.3%   404 2753s
 13481  8179    3.55344   35  124    6.08397    1.01527  83.3%   404 2828s
 14126  8304    1.01911   25  282    6.08397    1.01527  83.3%   404 2904s
 14554  8506    2.04902   27  321    6.08397    1.01527  83.3%   410 2977s
 14992  8790    1.13246   27  261    6.08397    1.01527  83.3%   416 3052s
 15542  9433    2.06344   37  195    6.08397    1.01527  83.3%   420 3133s
 16346 10008    5.06107   97   75    6.08397    1.01527  83.3%   419 3249s
 17193 10312 infeasible   30         6.08397    1.01527  83.3%   414 3335s
 17660 10362    6.06870   58   85    6.08397    1.01527  83.3%   424 3502s
 17717 11111    6.06870   63   78    6.08397    1.01527  83.3%   428 3600s

Cutting planes:
  Gomory: 6
  Cover: 10
  MIR: 5
  GUB cover: 32
  Zero half: 8
  RLT: 3

Explored 18620 nodes (8005872 simplex iterations) in 3600.87 seconds (5180.44 work units)
Thread count was 20 (of 20 available processors)

Solution count 10: 6.08397 6.0916 6.0916 ... 11.1298

Time limit reached
Best objective 6.083969465649e+00, best bound 1.015267175573e+00, gap 83.3124%
```

B.2 TALBP-1 MILP-2 Log

Gurobi log for P65 instance with $c = 490$.

```
Gurobi 12.0.3 (win64) logging started Mon Oct 13 19:16:12 2025

Set parameter OutputFlag to value 1
Set parameter TimeLimit to value 3600
Gurobi Optimizer version 12.0.3 build v12.0.3rc0 (win64 - Windows 11.0 (26100.2))

CPU model: 13th Gen Intel(R) Core(TM) i7-13700H, instruction set [SSE2|AVX|AVX2]
```

```
Thread count: 14 physical cores, 20 logical processors, using up to 20 threads

Non-default parameters:
TimeLimit  3600
MIPGap  0
LogToConsole  0

Optimize a model with 156221 rows, 9621 columns and 822378 nonzeros
Model fingerprint: 0xf4b5aa6c
Variable types: 65 continuous, 9556 integer (9556 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+03]
  Objective range  [8e-03, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 3e+03]
Presolve removed 71 rows and 1827 columns
Presolve time: 1.08s
Presolved: 156150 rows, 7794 columns, 805939 nonzeros
Variable types: 64 continuous, 7730 integer (7671 binary)
Root relaxation presolved: 7794 rows, 163944 columns, 813733 nonzeros

Root relaxation: objective 6.533898e-01, 12031 iterations, 2.44 seconds (4.05 work units)
Total elapsed time = 7.80s (DegenMoves)
Total elapsed time = 10.31s (DegenMoves)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.65339    0  347          -    0.65339      -     -   13s
     0     0    2.23942    0  493          -    2.23942      -     -   22s
     0     0    3.14632    0  471          -    3.14632      -     -   31s
     0     0    3.14632    0  463          -    3.14632      -     -   31s
     0     0    3.14634    0  466          -    3.14634      -     -   31s
     0     0    3.21068    0  507          -    3.21068      -     -   52s
     0     0    3.22687    0  486          -    3.22687      -     -   58s
     0     0    3.27304    0  617          -    3.27304      -     -   78s
     0     0    3.27317    0  602          -    3.27317      -     -   78s
     0     0    5.01863    0  524          -    5.01863      -     -   91s
H    0     0                      22.1526718    5.01863  77.3%     -  111s
H    0     0                      22.1374046    5.01863  77.3%     -  111s
H    0     0                      22.1221374    5.01863  77.3%     -  111s
H    0     0                      22.1145038    5.01863  77.3%     -  111s
     0     0    5.01863    0  378   22.11450    5.01863  77.3%     -  114s
     0     0    5.01863    0  379   22.11450    5.01863  77.3%     -  116s
     0     0    5.01863    0  487   22.11450    5.01863  77.3%     -  120s
     0     0    5.01863    0 1800   22.11450    5.01863  77.3%     -  123s
     0     0    5.01863    0 1045   22.11450    5.01863  77.3%     -  135s
     0     0    6.01796    0 1814   22.11450    6.01796  72.8%     -  142s
     0     0    6.04824    0  380   22.11450    6.04824  72.7%     -  143s
     0     0    6.04829    0  368   22.11450    6.04829  72.7%     -  143s
     0     0    6.07944    0  190   22.11450    6.07944  72.5%     -  146s
H    0     0                      22.1068702    6.07944  72.5%     -  146s
H    0     0                      22.0992366    6.07944  72.5%     -  146s
     0     0    6.07944    0  191   22.09924    6.07944  72.5%     -  146s
     0     0    6.07944    0  189   22.09924    6.07944  72.5%     -  147s
     0     0    6.07944    0  163   22.09924    6.07944  72.5%     -  147s
     0     0    6.07944    0   76   22.09924    6.07944  72.5%     -  149s
H    0     0                       9.0992366    6.07944  33.2%     -  149s
H    0     0                       6.0916031    6.07944  0.20%     -  149s
```

```
     0     0    6.07944    0  136    6.09160    6.07944  0.20%     -  150s
     0     0    6.07944    0  178    6.09160    6.07944  0.20%     -  150s
     0     0    6.07944    0  452    6.09160    6.07944  0.20%     -  151s
     0     0    6.07944    0  264    6.09160    6.07944  0.20%     -  155s
     0     0    6.07944    0  191    6.09160    6.07944  0.20%     -  155s
     0     0    6.07944    0  173    6.09160    6.07944  0.20%     -  155s
     0     0    6.07944    0  210    6.09160    6.07944  0.20%     -  155s
     0     0    6.07944    0  180    6.09160    6.07944  0.20%     -  155s
     0     0    6.07944    0  221    6.09160    6.07944  0.20%     -  156s
     0     0    6.07944    0  196    6.09160    6.07944  0.20%     -  156s
     0     0    6.07944    0   92    6.09160    6.07944  0.20%     -  158s
     0     0    6.07944    0  113    6.09160    6.07944  0.20%     -  158s
     0     0    6.07944    0   62    6.09160    6.07944  0.20%     -  158s
     0     0    6.07944    0  102    6.09160    6.07944  0.20%     -  159s
     0     0    6.07944    0   72    6.09160    6.07944  0.20%     -  159s
     0     0    6.07944    0  150    6.09160    6.07944  0.20%     -  159s
     0     0    6.07944    0  147    6.09160    6.07944  0.20%     -  159s
     0     0    6.07944    0   85    6.09160    6.07944  0.20%     -  160s
     0     0    6.07944    0  103    6.09160    6.07944  0.20%     -  160s
     0     0    6.07944    0   92    6.09160    6.07944  0.20%     -  161s
     0     0    6.07944    0  166    6.09160    6.07944  0.20%     -  161s
     0     0    6.07944    0   60    6.09160    6.07944  0.20%     -  161s
     0     0    6.07944    0   60    6.09160    6.07944  0.20%     -  162s
     0     2    6.07944    0   60    6.09160    6.07944  0.20%     -  162s
    71    26    6.07944    9  188    6.09160    6.07944  0.20%   317  165s
H  481   373                        6.0839695    6.07944  0.07%   177  168s

Cutting planes:
  Learned: 1
  Gomory: 8
  Cover: 547
  Implied bound: 104
  Clique: 32
  MIR: 255
  StrongCG: 5
  GUB cover: 109
  Zero half: 55
  Network: 4
  RLT: 53
  Relax-and-lift: 50
  BQP: 11
  PSD: 2

Explored 499 nodes (259814 simplex iterations) in 168.73 seconds (256.11 work units)
Thread count was 20 (of 20 available processors)

Solution count 9: 6.08397 6.0916 9.09924 ... 22.1527

Optimal solution found (tolerance 1.00e-04)
Best objective 6.083969465649e+00, best bound 6.083969465649e+00, gap 0.0000%
```

B.3 HRALBP-3 MILP Log

Gurobi log for Case Study instance with $n^h = 2$ and $n^r = 2$.

```
Gurobi 12.0.3 (win64) logging started Fri Oct 31 19:20:46 2025
```

```
Set parameter OutputFlag to value 1
Set parameter TimeLimit to value 3600
Gurobi Optimizer version 12.0.3 build v12.0.3rc0 (win64 - Windows 11.0 (26100.2))

CPU model: 13th Gen Intel(R) Core(TM) i7-13700H, instruction set [SSE2|AVX|AVX2]
Thread count: 14 physical cores, 20 logical processors, using up to 20 threads

Non-default parameters:
TimeLimit  3600
MIPGap  0
LogToConsole  0

Optimize a model with 195877 rows, 48804 columns and 728274 nonzeros
Model fingerprint: 0xecd36940
Variable types: 56 continuous, 48748 integer (48748 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+03]
  Objective range  [1e+00, 2e+03]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 3e+03]
Presolve removed 85113 rows and 29006 columns (presolve time = 5s)...
Presolve removed 85096 rows and 28989 columns
Presolve time: 5.14s
Presolved: 110781 rows, 19815 columns, 511130 nonzeros
Variable types: 26 continuous, 19789 integer (19748 binary)

Deterministic concurrent LP optimizer: primal and dual simplex (primal and dual model)
Showing primal log only...

Root relaxation presolved: 110781 rows, 19815 columns, 511130 nonzeros

Root simplex log...

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    0.0000000e+00   1.130625e+02   1.299586e+09      6s
Concurrent spin time: 0.10s

Solved with dual simplex

Root relaxation: objective 8.066654e+02, 1731 iterations, 0.76 seconds (0.84 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0  806.66540    0  258          -  806.66540      -     -   11s
     0     0 1883.61480    0  264          - 1883.61480      -     -   15s
H    0     0                    46663.720000 1883.61480  96.0%     -   15s
H    0     0                    11464.380000 1883.61480  83.6%     -   15s
H    0     0                    8061.3500000 1883.61480  76.6%     -   15s
H    0     0                    7559.2800000 1883.61480  75.1%     -   16s
H    0     0                    6413.0500000 1883.61480  70.6%     -   16s
H    0     0                    6411.6400000 1883.61480  70.6%     -   16s
H    0     0                    5910.5400000 1883.61480  68.1%     -   16s
H    0     0                    5909.2800000 1883.61480  68.1%     -   17s
H    0     0                    5559.2800000 1883.61480  66.1%     -   17s
H    0     0                    5555.9800000 1883.61480  66.1%     -   18s
H    0     0                    5205.9800000 1883.61480  63.8%     -   18s
H    0     0                    5053.7800000 1883.61480  62.7%     -   20s
H    0     0                    4553.7800000 1883.61480  58.6%     -   20s
```

```
       0     0 1913.64593    0  152 4553.78000 1913.64593  58.0%     -   23s
       0     0 1913.64593    0   99 4553.78000 1913.64593  58.0%     -   23s
       0     0 2353.78000    0  123 4553.78000 2353.78000  48.3%     -   23s
       0     0 2353.78000    0  123 4553.78000 2353.78000  48.3%     -   23s
       0     0 2353.78000    0   38 4553.78000 2353.78000  48.3%     -   24s
       0     0 2353.78000    0   34 4553.78000 2353.78000  48.3%     -   24s
       0     0 2353.78000    0   37 4553.78000 2353.78000  48.3%     -   24s
       0     0 2353.78000    0   35 4553.78000 2353.78000  48.3%     -   24s
       0     0 2353.78000    0   33 4553.78000 2353.78000  48.3%     -   24s
       0     0 2353.78000    0   40 4553.78000 2353.78000  48.3%     -   24s
       0     0 2353.78000    0   44 4553.78000 2353.78000  48.3%     -   24s
       0     0 2353.78000    0   45 4553.78000 2353.78000  48.3%     -   24s
       0     0 2353.78000    0   51 4553.78000 2353.78000  48.3%     -   24s
       0     0 2353.78000    0   53 4553.78000 2353.78000  48.3%     -   24s
       0     0 2353.78000    0   47 4553.78000 2353.78000  48.3%     -   24s
       0     0 2353.78000    0   48 4553.78000 2353.78000  48.3%     -   24s
       0     0 2353.78000    0   47 4553.78000 2353.78000  48.3%     -   24s
       0     0 2353.78000    0   47 4553.78000 2353.78000  48.3%     -   24s
       0     2 2353.83953    0   44 4553.78000 2353.83953  48.3%     -   25s
    1763   894 3734.81027   15  119 4553.78000 2354.35159  48.3%  37.4   30s
    3618  1533 4278.66281   11  109 4553.78000 2371.81656  47.9%  35.1   35s
    3828  1667 4206.81407   19   54 4553.78000 2371.81656  47.9%  41.1   40s
    4414  1746 2855.75998   21   47 4553.78000 2371.81656  47.9%  44.3   45s
    5078  1770 4372.09395   22   30 4553.78000 2371.81656  47.9%  47.9   50s
*  5999  1708              56    4509.9000000 2432.97810  46.1%  50.7   52s
    6979  1966     cutoff   40      4509.90000 2506.59979  44.4%  51.6   55s
    9618  2838     cutoff   41      4509.90000 2507.08344  44.4%  51.9   60s
   12254  3935     cutoff   26      4509.90000 2509.06624  44.4%  52.6   65s
   14939  4710     cutoff   31      4509.90000 2525.58508  44.0%  54.4   70s
   17688  6040 3207.78000   33   21 4509.90000 2703.77950  40.0%  53.4   75s
   19329  7242 3206.40704   30   29 4509.90000 2703.77950  40.0%  52.1   85s
   24871  8386 infeasible   39      4509.90000 2703.78000  40.0%  49.4   91s
   28567  9326 infeasible   46      4509.90000 2703.78000  40.0%  48.1   96s
   31045  9758     cutoff   35      4509.90000 2703.78000  40.0%  47.8  100s
   33645 10453     cutoff   34      4509.90000 2703.78000  40.0%  47.6  105s
   37381 11407 3205.10000   30   33 4509.90000 2703.78000  40.0%  47.6  111s
   38950 11408 3205.19000   31  152 4509.90000 2703.78000  40.0%  47.4  130s
   38963 11417 3207.26000   42   38 4509.90000 2703.78000  40.0%  47.4  135s
   38970 11421 3207.26000   41   61 4509.90000 2703.78000  40.0%  47.4  140s
   39003 11469 3703.82295   26  175 4509.90000 2703.78000  40.0%  47.9  145s
   39777 11607 4289.65622   31   52 4509.90000 2703.78000  40.0%  48.4  150s
   40745 11817 2856.60000   30   27 4509.90000 2703.78000  40.0%  48.5  155s
   42597 12121 infeasible   51      4509.90000 2703.78000  40.0%  48.7  160s
   45007 12094     cutoff   53      4509.90000 2703.78000  40.0%  48.6  165s
   48002 11968 3207.39000   50   22 4509.90000 2703.78000  40.0%  48.6  170s
   50195 12116 2856.82405   42   26 4509.90000 2703.78000  40.0%  48.7  175s
   53101 11865 4360.34101   62   44 4509.90000 2703.78000  40.0%  48.7  180s
   56221 11808     cutoff   40      4509.90000 2703.78000  40.0%  48.7  185s
   58854 11780     cutoff   51      4509.90000 2703.78000  40.0%  48.7  192s
   60744 11630     cutoff   53      4509.90000 2703.78000  40.0%  48.9  195s
   63777 11404 3196.58755   52   92 4509.90000 2703.78000  40.0%  49.0  201s
   66102 11417     cutoff   49      4509.90000 2703.78000  40.0%  49.0  205s
   68586 11250     cutoff   50      4509.90000 2703.78000  40.0%  48.8  210s
   71121 10974     cutoff   65      4509.90000 2703.78000  40.0%  48.8  215s
   73471 11011 2703.78000   39   21 4509.90000 2703.78000  40.0%  48.9  221s
   76429 11852 2856.85514   47   29 4509.90000 2703.78000  40.0%  49.0  226s
   78092 12151     cutoff   42      4509.90000 2703.78000  40.0%  48.9  231s
   81213 12839 3209.33000   56   16 4509.90000 2703.78000  40.0%  48.9  237s
   84264 13455 4206.76986   39   41 4509.90000 2703.78000  40.0%  49.0  242s
```

```
85800 13673 2742.68804   46   47 4509.90000 2703.78000  40.0%  49.0  245s
88920 14555 2858.72838   48   25 4509.90000 2703.78000  40.0%  49.0  251s
91846 15189 2858.67000   44   24 4509.90000 2703.78000  40.0%  49.0  257s
94799 15908     cutoff   49      4509.90000 2703.78000  40.0%  48.9  262s
96331 16233     cutoff   50      4509.90000 2703.78000  40.0%  48.9  265s
98900 16915 2856.67167   48   32 4509.90000 2703.78000  40.0%  49.0  272s
100633 17248 2713.52438   45   40 4509.90000 2703.78000  40.0%  49.1  276s
104038 18111 4226.56075   50   70 4509.90000 2703.78000  40.0%  49.1  283s
105814 18481 3207.12000   46   19 4509.90000 2703.78000  40.0%  49.0  286s
108601 19194     cutoff   43      4509.90000 2703.78000  40.0%  49.1  291s
111400 20030 4206.76986   38   47 4509.90000 2703.78000  40.0%  49.2  297s
113179 20299     cutoff   62      4509.90000 2703.78000  40.0%  49.2  301s
115892 20969 3205.29667   43   18 4509.90000 2703.78000  40.0%  49.3  306s
117236 21279     cutoff   43      4509.90000 2703.78000  40.0%  49.4  310s
120229 21812     cutoff   40      4509.90000 2703.78000  40.0%  49.5  316s
120974 22156 4358.95496   44   31 4509.90000 2703.78000  40.0%  49.6  323s
122219 22568 2857.53300   38   37 4509.90000 2703.78000  40.0%  49.6  327s
124111 22943     cutoff   56      4509.90000 2703.78000  40.0%  49.7  331s
126178 23142 3205.19000   49   19 4509.90000 2703.78000  40.0%  49.7  337s
127123 23419     cutoff   46      4509.90000 2703.78000  40.0%  49.7  341s
130153 23740 2858.85078   53   24 4509.90000 2703.87474  40.0%  49.8  345s
132371 24087 2856.92389   45   37 4509.90000 2704.15296  40.0%  49.9  351s
133868 24530     cutoff   47      4509.90000 2704.44000  40.0%  50.1  355s
137146 24936 2856.60000   45   20 4509.90000 2706.11393  40.0%  50.2  362s
138395 25097 2720.15599   41   37 4509.90000 2706.14000  40.0%  50.3  365s
141055 24915     cutoff   43      4509.90000 2710.82904  39.9%  50.6  371s
143268 24363 infeasible   45      4509.90000 2721.02480  39.7%  51.2  378s
144606 24279 4380.54000   41   49 4509.90000 2731.06619  39.4%  51.5  381s
146146 24316     cutoff   39      4509.90000 2747.16358  39.1%  51.7  385s
147881 24714 2856.60000   41   20 4509.90000 2782.75632  38.3%  51.9  390s
150328 25129 2943.39500   47   18 4509.90000 2856.60000  36.7%  51.8  395s
154674 26066 3203.78000   47   17 4509.90000 2856.60000  36.7%  51.5  403s
156862 26480 3211.30248   55   17 4509.90000 2856.60000  36.7%  51.4  408s
159033 26760 2858.88819   45   22 4509.90000 2856.60000  36.7%  51.2  411s
161725 27317 3205.19000   53   17 4509.90000 2856.60000  36.7%  51.1  416s
164484 27838 3205.85000   56   15 4509.90000 2856.60000  36.7%  51.0  421s
167183 28506 2857.68587   39   43 4509.90000 2856.60000  36.7%  50.9  427s
169255 28873     cutoff   55      4509.90000 2856.60000  36.7%  50.8  430s
171508 29429     cutoff   43      4509.90000 2856.60000  36.7%  50.8  435s
174423 29988     cutoff   54      4509.90000 2856.60000  36.7%  50.6  440s
177445 30764 3205.33392   42   26 4509.90000 2856.60000  36.7%  50.5  448s
179134 31105     cutoff   51      4509.90000 2856.60000  36.7%  50.5  451s
182046 31639     cutoff   46      4509.90000 2856.60000  36.7%  50.4  457s
183547 31954 2857.56500   40   37 4509.90000 2856.60000  36.7%  50.4  460s
186694 32559 4360.18691   51   32 4509.90000 2856.60000  36.7%  50.3  466s
189613 33092 3204.09543   52   16 4509.90000 2856.60000  36.7%  50.2  471s
192510 33795 infeasible   53      4509.90000 2856.60000  36.7%  50.1  477s
195137 34076     cutoff   56      4509.90000 2856.60000  36.7%  50.1  480s
196762 34764 3205.56413   55   24 4509.90000 2856.60000  36.7%  50.0  487s
199049 35083 3206.32000   45   23 4509.90000 2856.60000  36.7%  50.0  490s
201412 35784     cutoff   41      4509.90000 2856.60000  36.7%  49.9  495s
204249 36220 3204.82673   49   44 4509.90000 2856.60000  36.7%  49.8  501s
206478 36807 3205.19000   51   23 4509.90000 2856.60000  36.7%  49.8  507s
207992 37130 2860.06743   45   39 4509.90000 2856.60000  36.7%  49.8  511s
211225 37856 infeasible   61      4509.90000 2856.60000  36.7%  49.7  517s
212763 38176 3205.21777   50   23 4509.90000 2856.60000  36.7%  49.7  520s
215824 38890 3205.19000   51   21 4509.90000 2856.60000  36.7%  49.6  526s
217328 39320 3205.32356   57   39 4509.90000 2856.60000  36.7%  49.6  530s
220405 39945     cutoff   46      4509.90000 2856.60000  36.7%  49.5  536s
```

223413	40752	cutoff	62		4509.90000	2856.60000	36.7%	49.4	542s
225067	41023	2856.63409	50	32	4509.90000	2856.60000	36.7%	49.3	545s
228084	41718	cutoff	55		4509.90000	2856.60000	36.7%	49.2	551s
231217	42377	4371.11753	49	86	4509.90000	2856.60000	36.7%	49.1	557s
232737	42688	2857.18000	40	27	4509.90000	2856.60000	36.7%	49.1	560s
235756	43373	3207.40000	45	30	4509.90000	2856.60000	36.7%	48.9	566s
237301	43801	cutoff	51		4509.90000	2856.60000	36.7%	48.9	571s
240272	44168	3207.00500	54	30	4509.90000	2856.60901	36.7%	48.8	576s
242780	44525	2856.63409	46	30	4509.90000	2856.63409	36.7%	48.9	582s
244208	44704	3205.91140	55	27	4509.90000	2856.64353	36.7%	48.9	585s
246875	44909	infeasible	52		4509.90000	2856.66295	36.7%	49.0	592s
248224	44962	cutoff	65		4509.90000	2856.68509	36.7%	49.0	595s
251334	45438	cutoff	53		4509.90000	2856.71940	36.7%	49.0	601s
254320	45944	3207.92172	53	27	4509.90000	2856.75706	36.7%	49.0	607s
255787	46188	cutoff	41		4509.90000	2856.77667	36.7%	49.0	610s
258537	46408	3207.26000	49	28	4509.90000	2856.82405	36.7%	49.1	618s
260065	46566	infeasible	53		4509.90000	2856.83278	36.7%	49.1	622s
261663	46724	cutoff	51		4509.90000	2856.85881	36.7%	49.1	625s
264968	47116	cutoff	55		4509.90000	2856.96070	36.7%	49.2	632s
266386	47261	cutoff	47		4509.90000	2857.00333	36.7%	49.2	635s
269354	47734	cutoff	51		4509.90000	2857.11929	36.6%	49.3	643s
271013	48195	2860.04794	41	38	4509.90000	2857.18000	36.6%	49.2	646s
273074	48202	infeasible	46		4509.90000	2857.18000	36.6%	49.2	651s
273120	48225	3206.07927	52	20	4509.90000	2857.18000	36.6%	49.2	656s
273506	48512	cutoff	45		4509.90000	2857.18000	36.6%	49.2	662s
274703	48893	3207.92000	51	20	4509.90000	2857.18000	36.6%	49.1	665s
277407	49301	3207.16154	59	26	4509.90000	2857.18000	36.6%	49.0	670s
279802	49782	4358.67000	45	34	4509.90000	2857.22447	36.6%	49.0	678s
281445	49904	cutoff	49		4509.90000	2857.29321	36.6%	49.0	681s
283897	50214	infeasible	56		4509.90000	2857.40381	36.6%	49.0	687s
285319	50446	3032.51000	50	34	4509.90000	2857.49862	36.6%	49.0	691s
288521	50880	3207.26000	51	8	4509.90000	2857.70116	36.6%	49.0	697s
289736	51210	2867.20789	45	40	4509.90000	2857.76000	36.6%	49.0	701s
291589	51417	cutoff	50		4509.90000	2857.76000	36.6%	49.0	705s
294661	51795	cutoff	49		4509.90000	2857.76000	36.6%	49.0	711s
295850	52084	2884.81296	53	35	4509.90000	2857.81836	36.6%	49.0	715s
299307	52472	cutoff	45		4509.90000	2857.89144	36.6%	48.9	722s
300506	52672	2928.25200	48	30	4509.90000	2858.00645	36.6%	48.9	725s
303240	52947	cutoff	54		4509.90000	2858.08440	36.6%	48.9	732s
304652	53161	4358.67000	47	24	4509.90000	2858.31023	36.6%	49.0	735s
308193	53514	4488.96000	52	25	4509.90000	2858.53000	36.6%	48.9	743s
309988	53759	3207.15409	58	23	4509.90000	2858.53000	36.6%	48.9	747s
311784	53981	3209.22877	65	21	4509.90000	2858.53000	36.6%	48.8	750s
315516	54414	3207.78000	54	18	4509.90000	2858.53000	36.6%	48.7	756s
318089	54845	3205.85000	50	16	4509.90000	2858.67000	36.6%	48.7	764s
319902	54965	3205.85000	48	14	4509.90000	2858.67000	36.6%	48.7	767s
321368	55104	cutoff	46		4509.90000	2858.67000	36.6%	48.7	770s
323885	55260	3206.64601	55	31	4509.90000	2858.67000	36.6%	48.7	776s
326821	55556	3205.85000	45	22	4509.90000	2858.67000	36.6%	48.7	782s
328297	55679	4358.67000	40	33	4509.90000	2858.67000	36.6%	48.7	785s
331028	55874	infeasible	43		4509.90000	2858.67000	36.6%	48.7	791s
333935	56159	3209.19000	55	22	4509.90000	2858.67000	36.6%	48.7	797s
335345	56205	cutoff	60		4509.90000	2858.67000	36.6%	48.8	800s
338033	56560	infeasible	48		4509.90000	2858.67000	36.6%	48.8	808s
339566	56591	cutoff	49		4509.90000	2858.67000	36.6%	48.8	811s
342678	56884	2876.47607	47	39	4509.90000	2858.67000	36.6%	48.8	817s
344172	57013	3207.92000	51	21	4509.90000	2858.67000	36.6%	48.8	821s
347212	57439	4371.75660	45	56	4509.90000	2858.67000	36.6%	48.7	827s
348754	57562	cutoff	54		4509.90000	2858.67000	36.6%	48.7	831s

```
351298 57632 4368.87280   45   54 4509.90000 2858.75013  36.6%  48.8  836s
353564 57925     cutoff   50      4509.90000 2858.80000  36.6%  48.9  842s
355011 58009 3208.97056   58   24 4509.90000 2858.83884  36.6%  48.9  845s
357719 58146 2867.43532   56  107 4509.90000 2859.08060  36.6%  48.9  853s
358972 58258     cutoff   55      4509.90000 2859.20727  36.6%  48.9  857s
360751 58329 3207.12000   49   22 4509.90000 2859.37800  36.6%  49.0  860s
363882 58655 infeasible   55      4509.90000 2859.81393  36.6%  49.0  867s
365321 58889 3208.61534   56   22 4509.90000 2859.90000  36.6%  49.0  870s
368385 59163 4381.34204   45   39 4509.90000 2860.56278  36.6%  49.0  877s
369883 59299 3207.78000   42   27 4509.90000 2860.74000  36.6%  49.0  881s
372755 59723     cutoff   54      4509.90000 2863.11256  36.5%  49.0  888s
374380 59998 3209.19681   59   21 4509.90000 2864.79108  36.5%  49.0  891s
376090 60293     cutoff   47      4509.90000 2865.56810  36.5%  48.9  895s
379269 60953 3209.53135   60   29 4509.90000 2867.05064  36.4%  48.9  903s
380736 61371 3205.85000   58   24 4509.90000 2867.78699  36.4%  48.8  906s
383914 62092     cutoff   59      4509.90000 2868.45235  36.4%  48.8  912s
385591 62369 infeasible   52      4509.90000 2868.75051  36.4%  48.7  915s
388360 62894 2907.98484   54   75 4509.90000 2869.23136  36.4%  48.7  921s
391083 63340     cutoff   52      4509.90000 2870.47163  36.4%  48.6  927s
392626 63614 3207.60688   52   28 4509.90000 2870.66927  36.3%  48.6  931s
395371 64133 2875.93239   51   40 4509.90000 2871.57110  36.3%  48.6  936s
396769 64443 3204.38924   45   36 4509.90000 2871.66784  36.3%  48.6  942s
398394 64745     cutoff   54      4509.90000 2872.10916  36.3%  48.5  945s
401173 65039     cutoff   50      4509.90000 2872.53530  36.3%  48.5  951s
403937 65438 4356.60000   44   34 4509.90000 2874.23763  36.3%  48.5  957s
405327 65641 4372.23671   56   92 4509.90000 2874.94470  36.3%  48.5  960s
407861 66021 3039.68938   47   47 4509.90000 2875.82419  36.2%  48.5  967s
409319 66157     cutoff   52      4509.90000 2876.74386  36.2%  48.5  970s
412268 66638 3203.78000   47   23 4509.90000 2878.54371  36.2%  48.5  976s
413739 66799     cutoff   57      4509.90000 2879.49520  36.2%  48.5  980s
416240 66978 4379.27598   41   79 4509.90000 2881.56520  36.1%  48.5  987s
417645 67040     cutoff   51      4509.90000 2882.82131  36.1%  48.5  990s
420612 67307 3205.85000   47   23 4509.90000 2886.61398  36.0%  48.6  997s
421988 67328     cutoff   42      4509.90000 2889.27873  35.9%  48.6 1001s
424794 67433 infeasible   50      4509.90000 2895.82522  35.8%  48.6 1007s
426064 67548     cutoff   49      4509.90000 2899.19790  35.7%  48.7 1010s
429067 67631     cutoff   45      4509.90000 2907.91872  35.5%  48.7 1017s
430345 67621 3209.19000   67   12 4509.90000 2913.23731  35.4%  48.7 1020s
433074 67631 infeasible   46      4509.90000 2928.00600  35.1%  48.8 1026s
435649 67182     cutoff   50      4509.90000 2950.69667  34.6%  48.9 1033s
436930 67027     cutoff   44      4509.90000 2963.76533  34.3%  49.0 1036s
438444 67018     cutoff   57      4509.90000 2994.50854  33.6%  49.1 1040s
441196 66446     cutoff   45      4509.90000 3042.10744  32.5%  49.1 1047s
442802 66735     cutoff   72      4509.90000 3139.87933  30.4%  49.2 1051s
444954 67078 3207.52958   55   22 4509.90000 3203.77984  29.0%  49.1 1057s
446747 67298 3204.44942   49   32 4509.90000 3203.78000  29.0%  49.0 1060s
449948 67964 infeasible   62      4509.90000 3203.78000  29.0%  49.0 1067s
451490 68176 3204.48500   53   28 4509.90000 3203.78000  29.0%  48.9 1070s
454523 68739     cutoff   64      4509.90000 3203.78000  29.0%  48.9 1076s
457246 69268 infeasible   50      4509.90000 3203.78000  29.0%  48.8 1082s
458711 69549 3207.26000   55   16 4509.90000 3203.78000  29.0%  48.8 1085s
461685 70026 infeasible   49      4509.90000 3203.78000  29.0%  48.7 1092s
463197 70285 3203.84962   49   33 4509.90000 3203.78000  29.0%  48.7 1095s
464630 70651 3207.26000   52   22 4509.90000 3203.78000  29.0%  48.6 1101s
468099 71084 3205.19000   43   26 4509.90000 3203.78000  29.0%  48.6 1107s
469240 71369 3205.13557   46   24 4509.90000 3203.78000  29.0%  48.5 1110s
472326 72041 3204.58189   48   30 4509.90000 3203.78000  29.0%  48.5 1116s
475389 72551 3207.43534   56   21 4509.90000 3203.78000  29.0%  48.4 1122s
476806 72848 3205.98919   53   27 4509.90000 3203.78000  29.0%  48.4 1126s
```

479797	73299	3204.13250	46	33	4509.90000	3203.78000	29.0%	48.3	1132s
480956	73572	3204.13250	47	32	4509.90000	3203.78000	29.0%	48.3	1136s
483848	74066	4176.16762	51	31	4509.90000	3203.78000	29.0%	48.2	1144s
485637	74391	3205.85000	46	15	4509.90000	3203.78000	29.0%	48.2	1148s
487156	74706	cutoff	49		4509.90000	3203.78000	29.0%	48.1	1151s
490171	75211	3207.26000	45	19	4509.90000	3203.78000	29.0%	48.1	1158s
491620	75388	3203.78000	44	29	4509.90000	3203.78000	29.0%	48.0	1161s
494489	75917	3205.19000	48	17	4509.90000	3203.78000	29.0%	48.0	1168s
496078	76185	3208.42000	54	20	4509.90000	3203.78000	29.0%	48.0	1171s
497593	76378	infeasible	46		4509.90000	3203.78000	29.0%	47.9	1175s
500371	76813	cutoff	66		4509.90000	3203.78000	29.0%	47.9	1182s
501756	77209	cutoff	53		4509.90000	3203.78000	29.0%	47.9	1185s
503450	77525	3206.61640	51	25	4509.90000	3203.78000	29.0%	47.8	1191s
506869	78154	3203.84607	54	25	4509.90000	3203.78000	29.0%	47.7	1197s
508296	78470	3205.19000	44	27	4509.90000	3203.78000	29.0%	47.7	1201s
510733	78927	3206.27265	52	31	4509.90000	3203.78000	29.0%	47.7	1208s
512255	79140	3209.33000	58	16	4509.90000	3203.78000	29.0%	47.6	1211s
513632	79440	3203.78000	45	30	4509.90000	3203.78000	29.0%	47.6	1215s
516591	79905	3207.26000	56	23	4509.90000	3203.78000	29.0%	47.6	1222s
518034	80178	3203.87397	49	35	4509.90000	3203.78000	29.0%	47.5	1225s
521066	80730	cutoff	57		4509.90000	3203.78000	29.0%	47.5	1232s
522465	80982	3205.77472	47	28	4509.90000	3203.78000	29.0%	47.5	1235s
523905	81281	3203.78000	46	23	4509.90000	3203.78000	29.0%	47.5	1241s
527107	81661	cutoff	49		4509.90000	3203.78000	29.0%	47.4	1248s
528450	81868	3203.78000	45	34	4509.90000	3203.78000	29.0%	47.4	1251s
531237	82261	3204.13678	50	40	4509.90000	3203.78000	29.0%	47.4	1258s
532462	82453	cutoff	52		4509.90000	3203.78000	29.0%	47.4	1262s
533969	82711	3207.92000	54	30	4509.90000	3203.78000	29.0%	47.4	1266s
535527	82946	3205.85000	53	25	4509.90000	3203.78000	29.0%	47.4	1270s
538725	83386	3204.94000	46	30	4509.90000	3203.78000	29.0%	47.4	1277s
540178	83520	3203.82759	41	37	4509.90000	3203.78000	29.0%	47.4	1281s
541388	83798	3208.74581	56	34	4509.90000	3203.78000	29.0%	47.4	1285s
543058	84044	3203.82037	49	25	4509.90000	3203.78000	29.0%	47.3	1290s
546398	84348	3204.49186	52	54	4509.90000	3203.80241	29.0%	47.3	1298s
547867	84580	3209.33000	60	21	4509.90000	3203.81112	29.0%	47.2	1302s
549348	84669	infeasible	59		4509.90000	3203.81562	29.0%	47.2	1305s
551807	85038	cutoff	51		4509.90000	3203.82419	29.0%	47.2	1312s
553393	85261	3207.51219	56	30	4509.90000	3203.83358	29.0%	47.2	1316s
556490	85704	3209.35967	59	26	4509.90000	3203.84722	29.0%	47.1	1323s
558075	85771	3207.57258	53	24	4509.90000	3203.85542	29.0%	47.1	1327s
559339	86024	cutoff	53		4509.90000	3203.86203	29.0%	47.1	1330s
562494	86524	cutoff	57		4509.90000	3203.88371	29.0%	47.0	1339s
564284	86656	3204.02667	43	30	4509.90000	3203.88667	29.0%	47.0	1343s
565683	86877	3206.89994	56	25	4509.90000	3203.88667	29.0%	47.0	1346s
567143	86965	3206.71154	56	30	4509.90000	3203.88667	29.0%	47.0	1350s
570207	87356	cutoff	58		4509.90000	3203.91977	29.0%	46.9	1357s
571447	87560	3205.19000	46	26	4509.90000	3203.93289	29.0%	46.9	1361s
572985	87817	3205.85000	44	18	4509.90000	3203.94611	29.0%	46.9	1365s
576339	88242	cutoff	55		4509.90000	3203.95625	29.0%	46.8	1372s
577601	88395	3204.42208	52	27	4509.90000	3203.96262	29.0%	46.8	1375s
580278	88674	3208.46720	50	24	4509.90000	3203.99925	29.0%	46.8	1382s
581703	88881	3204.15843	53	30	4509.90000	3204.02273	29.0%	46.8	1386s
583261	89017	cutoff	52		4509.90000	3204.02667	29.0%	46.8	1390s
586017	89189	3214.39137	59	58	4509.90000	3204.04971	29.0%	46.8	1398s
587471	89279	3270.96553	45	44	4509.90000	3204.07000	29.0%	46.7	1402s
588972	89571	3204.13250	40	29	4509.90000	3204.10323	29.0%	46.7	1405s
590692	89939	cutoff	50		4509.90000	3204.12918	29.0%	46.7	1410s
594383	90659	3211.22000	59	16	4509.90000	3204.13250	29.0%	46.6	1418s
596007	91000	cutoff	61		4509.90000	3204.13250	29.0%	46.6	1422s

```
597737 91310      cutoff   59      4509.90000 3204.13250  29.0%  46.6 1426s
599176 91499      cutoff   48      4509.90000 3204.13250  29.0%  46.6 1430s
602243 91984 3207.04578   41   55 4509.90000 3204.13250  29.0%  46.5 1439s
604038 92057 3204.25775   46   41 4509.90000 3204.13250  29.0%  46.5 1443s
605229 92134 3204.24000   53   27 4509.90000 3204.15333  29.0%  46.5 1446s
606404 92214      cutoff   55      4509.90000 3204.15913  29.0%  46.5 1450s
609514 92566 3206.03506   53   22 4509.90000 3204.20197  29.0%  46.5 1458s
610987 92861 infeasible   56      4509.90000 3204.22000  29.0%  46.5 1462s
612666 93034 3207.50667   56   20 4509.90000 3204.24126  29.0%  46.4 1465s
615438 93155 3205.93443   56   23 4509.90000 3204.28708  28.9%  46.4 1473s
616705 93180 3205.26092   58   30 4509.90000 3204.31993  28.9%  46.4 1476s
618122 93291      cutoff   54      4509.90000 3204.34610  28.9%  46.4 1481s
620610 93559 3207.78000   49   30 4509.90000 3204.37867  28.9%  46.4 1488s
622054 93698 infeasible   42      4509.90000 3204.43213  28.9%  46.4 1493s
623466 93958 3205.19000   39   19 4509.90000 3204.47008  28.9%  46.4 1498s
625371 94202 3209.39369   54   17 4509.90000 3204.48500  28.9%  46.4 1501s
626979 94401 3205.08196   57   32 4509.90000 3204.48500  28.9%  46.4 1505s
629938 94857      cutoff   43      4509.90000 3204.48500  28.9%  46.3 1512s
631466 95170 3205.19000   44   21 4509.90000 3204.48500  28.9%  46.3 1516s
633034 95409 3205.19000   41   22 4509.90000 3204.48500  28.9%  46.3 1520s
636158 95959 3204.48500   46   30 4509.90000 3204.48500  28.9%  46.3 1527s
637515 96212 3204.55434   54   30 4509.90000 3204.48500  28.9%  46.2 1530s
639926 96422      cutoff   51      4509.90000 3204.49155  28.9%  46.2 1537s
641465 96505      cutoff   56      4509.90000 3204.51644  28.9%  46.2 1543s
642865 96570 infeasible   50      4509.90000 3204.53063  28.9%  46.2 1547s
644390 96672 3208.50241   58   17 4509.90000 3204.55672  28.9%  46.2 1551s
645972 96948 3205.71000   54   28 4509.90000 3204.57767  28.9%  46.2 1555s
649178 97142 3208.59060   56   29 4509.90000 3204.61496  28.9%  46.2 1563s
650600 97173      cutoff   51      4509.90000 3204.64024  28.9%  46.2 1567s
651946 97238 3205.60204   53   33 4509.90000 3204.67243  28.9%  46.2 1570s
654659 97330 3205.20634   54   31 4509.90000 3204.74500  28.9%  46.2 1578s
656140 97418 3205.85000   50   20 4509.90000 3204.79482  28.9%  46.2 1581s
657573 97440      cutoff   46      4509.90000 3204.82437  28.9%  46.2 1585s
660298 97415      cutoff   72      4509.90000 3204.90109  28.9%  46.2 1593s
661513 97355 3207.01000   40   19 4509.90000 3204.94000  28.9%  46.2 1597s
663149 97337 infeasible   35      4509.90000 3204.94000  28.9%  46.2 1602s
664681 97375      cutoff   56      4509.90000 3204.94000  28.9%  46.2 1607s
666484 97437      cutoff   51      4509.90000 3204.94000  28.9%  46.2 1613s
668104 97535 3206.58599   49   34 4509.90000 3204.94000  28.9%  46.2 1617s
669437 97688 3207.12000   63   19 4509.90000 3204.94000  28.9%  46.2 1621s
671003 97732 3205.19000   51   20 4509.90000 3204.94000  28.9%  46.2 1626s
672337 97790 3205.58500   52   17 4509.90000 3204.94000  28.9%  46.2 1630s
675168 97950      cutoff   53      4509.90000 3205.02533  28.9%  46.2 1637s
676682 97923 3210.59000   60   19 4509.90000 3205.05875  28.9%  46.2 1641s
677968 98019      cutoff   48      4509.90000 3205.11587  28.9%  46.2 1645s
680793 98480      cutoff   55      4509.90000 3205.17829  28.9%  46.2 1652s
682588 98821 3205.19000   44   23 4509.90000 3205.18951  28.9%  46.2 1657s
684344 99085 infeasible   52      4509.90000 3205.19000  28.9%  46.1 1661s
686018 99339      cutoff   44      4509.90000 3205.19000  28.9%  46.1 1666s
687714 99556      cutoff   53      4509.90000 3205.19000  28.9%  46.1 1670s
690692 99971 3205.19000   53   25 4509.90000 3205.19000  28.9%  46.1 1678s
692257 100223 3209.31633   61   21 4509.90000 3205.19000  28.9%  46.1 1682s
693855 100482 3207.12000   46   17 4509.90000 3205.19000  28.9%  46.0 1686s
697083 100915 3205.19000   58   22 4509.90000 3205.19000  28.9%  46.0 1693s
698496 101055 3207.26000   56   26 4509.90000 3205.19000  28.9%  46.0 1696s
699706 101238      cutoff   55      4509.90000 3205.19000  28.9%  46.0 1700s
702677 101761 3207.25042   57   23 4509.90000 3205.19000  28.9%  45.9 1707s
704191 101860      cutoff   58      4509.90000 3205.19000  28.9%  45.9 1710s
705130 102117 3207.26000   47   15 4509.90000 3205.19000  28.9%  45.9 1716s
```

```
707005 102332     cutoff   63      4509.90000 3205.19000  28.9%  45.9 1720s
709848 102762 3205.19000   54   15 4509.90000 3205.19000  28.9%  45.9 1728s
711301 102955     cutoff   59      4509.90000 3205.19000  28.9%  45.9 1732s
712622 103038     cutoff   55      4509.90000 3205.19000  28.9%  45.8 1736s
714118 103279 3207.33207   47   29 4509.90000 3205.19000  28.9%  45.8 1740s
717134 103708 infeasible   51      4509.90000 3205.19000  28.9%  45.8 1748s
718573 103999 3207.26000   54   21 4509.90000 3205.19000  28.9%  45.8 1752s
719981 104261     cutoff   64      4509.90000 3205.19000  28.9%  45.8 1756s
722844 104608     cutoff   52      4509.90000 3205.19000  28.9%  45.8 1764s
724340 104835 3209.19000   68   15 4509.90000 3205.19000  28.9%  45.7 1770s
726009 105096 3205.19000   54   26 4509.90000 3205.19000  28.9%  45.7 1775s
729105 105460 3209.33000   62   20 4509.90000 3205.19000  28.9%  45.7 1783s
730362 105725     cutoff   52      4509.90000 3205.19000  28.9%  45.7 1787s
731937 105977 infeasible   72      4509.90000 3205.19000  28.9%  45.7 1791s
733534 106197     cutoff   62      4509.90000 3205.19000  28.9%  45.7 1795s
736448 106642     cutoff   57      4509.90000 3205.19000  28.9%  45.6 1802s
737947 106865     cutoff   52      4509.90000 3205.19000  28.9%  45.6 1806s
739435 107071 3207.12000   44   25 4509.90000 3205.19000  28.9%  45.6 1810s
742024 107457     cutoff   72      4509.90000 3205.19000  28.9%  45.6 1817s
743448 107690 3209.33000   57   17 4509.90000 3205.19000  28.9%  45.6 1823s
745095 107905     cutoff   53      4509.90000 3205.19000  28.9%  45.6 1827s
746536 108108     cutoff   58      4509.90000 3205.19000  28.9%  45.6 1830s
749401 108639 infeasible   43      4509.90000 3205.19000  28.9%  45.5 1838s
751028 108866 3207.12000   59   15 4509.90000 3205.19000  28.9%  45.5 1843s
752418 109094 3206.13680   51   35 4509.90000 3205.19000  28.9%  45.5 1847s
753815 109293     cutoff   52      4509.90000 3205.19000  28.9%  45.5 1851s
755383 109469     cutoff   56      4509.90000 3205.19000  28.9%  45.5 1855s
758473 109902 3205.24119   53   36 4509.90000 3205.19000  28.9%  45.5 1862s
759902 110133 3209.19000   59   15 4509.90000 3205.19000  28.9%  45.5 1866s
761475 110380     cutoff   58      4509.90000 3205.19000  28.9%  45.4 1870s
762994 110603 3207.77367   54   25 4509.90000 3205.19000  28.9%  45.4 1876s
764949 110787     cutoff   61      4509.90000 3205.19000  28.9%  45.4 1880s
767520 111271     cutoff   53      4509.90000 3205.19000  28.9%  45.4 1888s
769063 111536     cutoff   60      4509.90000 3205.19000  28.9%  45.4 1892s
770827 111765     cutoff   47      4509.90000 3205.19000  28.9%  45.4 1896s
772457 111955 3207.39000   53   23 4509.90000 3205.19000  28.9%  45.4 1900s
775479 112420 3205.19000   46   19 4509.90000 3205.19000  28.9%  45.3 1908s
777002 112596 3207.63421   58   32 4509.90000 3205.19000  28.9%  45.3 1912s
778278 112851 3207.12000   58   20 4509.90000 3205.19000  28.9%  45.3 1917s
779917 113175 3207.26000   56   15 4509.90000 3205.19000  28.9%  45.3 1920s
783076 113756 3209.33000   54   20 4509.90000 3205.19000  28.9%  45.3 1930s
786251 114247     cutoff   58      4509.90000 3205.19000  28.9%  45.2 1937s
787762 114486 3205.70750   56   21 4509.90000 3205.19000  28.9%  45.2 1941s
790638 114867     cutoff   60      4509.90000 3205.19000  28.9%  45.2 1948s
792064 115080 infeasible   48      4509.90000 3205.19000  28.9%  45.2 1952s
793603 115327     cutoff   66      4509.90000 3205.19000  28.9%  45.2 1956s
795178 115504     cutoff   58      4509.90000 3205.19000  28.9%  45.2 1960s
798031 115977 3205.31179   44   29 4509.90000 3205.19000  28.9%  45.1 1967s
799586 116167 3205.19000   44   20 4509.90000 3205.19000  28.9%  45.1 1972s
800774 116357     cutoff   70      4509.90000 3205.19000  28.9%  45.1 1975s
802205 116622     cutoff   51      4509.90000 3205.19000  28.9%  45.1 1981s
803920 116851 3209.33000   73   16 4509.90000 3205.19000  28.9%  45.1 1985s
806867 117282 3205.19000   47   25 4509.90000 3205.19000  28.9%  45.1 1992s
808206 117488 3205.25414   56   23 4509.90000 3205.19000  28.9%  45.1 1996s
809614 117653     cutoff   67      4509.90000 3205.19000  28.9%  45.1 2000s
812403 118056     cutoff   54      4509.90000 3205.19000  28.9%  45.1 2008s
814023 118214 3209.33000   70   15 4509.90000 3205.19000  28.9%  45.0 2012s
815435 118428 3207.12000   57   22 4509.90000 3205.19000  28.9%  45.0 2016s
816924 118663     cutoff   70      4509.90000 3205.19000  28.9%  45.0 2020s
```

818484	118755	infeasible	55		4509.90000	3205.19000	28.9%	45.0	2025s
821328	119348	cutoff	49		4509.90000	3205.19000	28.9%	45.0	2034s
822991	119517	cutoff	53		4509.90000	3205.19000	28.9%	45.0	2038s
824271	119709	3207.44203	56	32	4509.90000	3205.19000	28.9%	45.0	2042s
825809	119976	3205.45845	47	34	4509.90000	3205.19000	28.9%	45.0	2046s
827506	120168	cutoff	70		4509.90000	3205.19000	28.9%	45.0	2051s
829042	120316	infeasible	61		4509.90000	3205.19000	28.9%	45.0	2055s
832034	120634	3207.32435	53	33	4509.90000	3205.19000	28.9%	44.9	2063s
833396	120902	3205.31664	36	34	4509.90000	3205.19000	28.9%	44.9	2067s
835012	121067	cutoff	62		4509.90000	3205.19000	28.9%	44.9	2071s
836493	121277	3205.19000	38	23	4509.90000	3205.19000	28.9%	44.9	2075s
839371	121639	cutoff	50		4509.90000	3205.19000	28.9%	44.9	2083s
840843	121868	3209.33000	56	21	4509.90000	3205.19000	28.9%	44.9	2089s
842568	122018	cutoff	56		4509.90000	3205.19000	28.9%	44.9	2093s
843812	122244	infeasible	54		4509.90000	3205.19000	28.9%	44.9	2097s
845387	122498	3205.43220	45	26	4509.90000	3205.19000	28.9%	44.9	2101s
848592	123134	3207.27946	56	25	4509.90000	3205.19000	28.9%	44.8	2108s
850219	123406	3207.26000	55	23	4509.90000	3205.19000	28.9%	44.8	2112s
851817	123659	3205.19000	48	21	4509.90000	3205.19000	28.9%	44.8	2116s
853348	123865	3205.19000	41	35	4509.90000	3205.19000	28.9%	44.8	2120s
856166	124312	3205.19000	56	22	4509.90000	3205.19000	28.9%	44.8	2128s
857726	124510	infeasible	37		4509.90000	3205.19000	28.9%	44.7	2131s
859020	124670	3207.26000	53	23	4509.90000	3205.19000	28.9%	44.7	2135s
860484	124884	3205.19000	41	36	4509.90000	3205.19000	28.9%	44.7	2141s
862707	125126	3205.19000	56	15	4509.90000	3205.19000	28.9%	44.7	2145s
865160	125456	cutoff	45		4509.90000	3205.19000	28.9%	44.7	2152s
866592	125649	3205.19000	53	19	4509.90000	3205.19000	28.9%	44.7	2155s
869359	125996	3205.26345	43	23	4509.90000	3205.19000	28.9%	44.7	2163s
870761	126178	3207.38336	59	33	4509.90000	3205.19000	28.9%	44.7	2167s
872187	126387	3205.19000	53	23	4509.90000	3205.19000	28.9%	44.6	2171s
873676	126753	3205.19000	56	21	4509.90000	3205.19000	28.9%	44.6	2175s
876737	127083	3209.37215	62	19	4509.90000	3205.19000	28.9%	44.6	2182s
878144	127299	cutoff	51		4509.90000	3205.19000	28.9%	44.6	2186s
879685	127541	3219.54313	60	40	4509.90000	3205.19000	28.9%	44.6	2192s
881384	127767	3209.29168	64	17	4509.90000	3205.19000	28.9%	44.6	2196s
882796	128026	3205.28990	68	26	4509.90000	3205.19000	28.9%	44.6	2200s
885871	128511	3207.26000	50	18	4509.90000	3205.19000	28.9%	44.5	2208s
887383	128777	3209.33000	59	17	4509.90000	3205.19000	28.9%	44.5	2212s
889048	128842	3207.46456	68	21	4509.90000	3205.19000	28.9%	44.5	2215s
890379	129227	3210.65000	58	20	4509.90000	3205.19000	28.9%	44.5	2222s
891942	129465	3209.19000	66	18	4509.90000	3205.19000	28.9%	44.5	2226s
893539	129651	3207.26000	52	25	4509.90000	3205.19000	28.9%	44.5	2230s
896292	130080	3207.26000	57	21	4509.90000	3205.19000	28.9%	44.5	2238s
897788	130318	3207.39000	47	19	4509.90000	3205.19000	28.9%	44.4	2242s
899285	130573	3205.70750	55	29	4509.90000	3205.19000	28.9%	44.4	2248s
900906	130793	cutoff	56		4509.90000	3205.19000	28.9%	44.4	2251s
902436	131018	3205.30079	56	26	4509.90000	3205.19000	28.9%	44.4	2255s
905398	131362	cutoff	66		4509.90000	3205.19000	28.9%	44.4	2262s
906709	131539	3208.58651	56	20	4509.90000	3205.19000	28.9%	44.4	2266s
908046	131698	3205.32143	56	34	4509.90000	3205.19000	28.9%	44.4	2270s
911008	132091	3207.26000	52	17	4509.90000	3205.19000	28.9%	44.4	2278s
912488	132338	cutoff	42		4509.90000	3205.19000	28.9%	44.4	2281s
913979	132570	cutoff	54		4509.90000	3205.19000	28.9%	44.4	2285s
916754	132892	3207.26000	52	15	4509.90000	3205.19000	28.9%	44.4	2293s
917959	133093	3205.26199	56	28	4509.90000	3205.19000	28.9%	44.4	2297s
919492	133253	cutoff	50		4509.90000	3205.19000	28.9%	44.4	2302s
921059	133445	3207.70210	58	19	4509.90000	3205.19000	28.9%	44.3	2306s
922699	133642	3207.26000	54	20	4509.90000	3205.19000	28.9%	44.3	2310s
924110	133838	3207.22108	53	28	4509.90000	3205.19000	28.9%	44.3	2315s

```
927121 134367     cutoff   67       4509.90000 3205.19000  28.9%  44.3 2323s
928683 134586 3205.19000   57   20 4509.90000 3205.19000  28.9%  44.3 2327s
930079 134800     cutoff   64       4509.90000 3205.19000  28.9%  44.3 2331s
931417 135022 3209.41400   64   20 4509.90000 3205.19000  28.9%  44.3 2335s
934314 135360 3206.99766   53   40 4509.90000 3205.19000  28.9%  44.3 2342s
935660 135570     cutoff   55       4509.90000 3205.19000  28.9%  44.3 2347s
937128 135778 3207.26000   54   19 4509.90000 3205.19000  28.9%  44.3 2351s
938577 136013 3205.40048   60   38 4509.90000 3205.19000  28.9%  44.3 2356s
940163 136259     cutoff   70       4509.90000 3205.19000  28.9%  44.3 2360s
941601 136400     cutoff   52       4509.90000 3205.19000  28.9%  44.3 2365s
943332 136651 3205.22399   62   38 4509.90000 3205.19000  28.9%  44.3 2370s
946141 137185 3205.19000   50   18 4509.90000 3205.19000  28.9%  44.2 2378s
947715 137373 3211.26000   67   13 4509.90000 3205.19000  28.9%  44.2 2382s
949109 137580 3207.41147   55   24 4509.90000 3205.19000  28.9%  44.2 2387s
950727 137760     cutoff   65       4509.90000 3205.19000  28.9%  44.2 2391s
952303 138024 3205.19000   53   15 4509.90000 3205.19000  28.9%  44.2 2395s
954696 138391     cutoff   55       4509.90000 3205.19000  28.9%  44.2 2404s
956260 138574 3205.30541   53   29 4509.90000 3205.19000  28.9%  44.2 2409s
957760 138808 3205.60754   57   28 4509.90000 3205.19000  28.9%  44.2 2414s
959351 139018     cutoff   51       4509.90000 3205.19000  28.9%  44.2 2419s
960886 139195     cutoff   52       4509.90000 3205.19000  28.9%  44.2 2423s
962290 139400 3208.58369   63   21 4509.90000 3205.19000  28.9%  44.2 2428s
963667 139501 3205.42119   41   27 4509.90000 3205.19000  28.9%  44.2 2432s
965034 139645 3206.26640   61   28 4509.90000 3205.19000  28.9%  44.2 2436s
966476 139845     cutoff   47       4509.90000 3205.19000  28.9%  44.2 2441s
968071 139962     cutoff   65       4509.90000 3205.19000  28.9%  44.1 2445s
969390 140162 3210.72771   57   17 4509.90000 3205.19000  28.9%  44.1 2450s
972462 140566 3205.19000   51   15 4509.90000 3205.19000  28.9%  44.1 2458s
973994 140738     cutoff   63       4509.90000 3205.19000  28.9%  44.1 2463s
975510 140929 3205.19000   51   29 4509.90000 3205.19000  28.9%  44.1 2467s
976801 141045     cutoff   68       4509.90000 3205.19000  28.9%  44.1 2472s
978211 141233 3205.19000   40   30 4509.90000 3205.19000  28.9%  44.1 2477s
979758 141419 3209.33000   59   18 4509.90000 3205.19000  28.9%  44.1 2482s
981396 141578     cutoff   46       4509.90000 3205.19000  28.9%  44.1 2486s
982867 141705     cutoff   59       4509.90000 3205.19000  28.9%  44.1 2490s
985696 141918 3205.19000   47   22 4509.90000 3205.19000  28.9%  44.1 2495s
987220 142260 3207.26000   52   21 4509.90000 3205.19000  28.9%  44.1 2503s
988708 142426     cutoff   51       4509.90000 3205.19000  28.9%  44.1 2507s
990144 142586 3207.48314   51   26 4509.90000 3205.19000  28.9%  44.1 2511s
991603 142747 3207.26000   58   19 4509.90000 3205.19000  28.9%  44.1 2515s
993131 142937 3207.17169   63   21 4509.90000 3205.19000  28.9%  44.1 2520s
996269 143375     cutoff   41       4509.90000 3205.19000  28.9%  44.1 2529s
997807 143599 3207.26000   53   23 4509.90000 3205.19000  28.9%  44.1 2535s
1000823 144021 3207.12000   46   26 4509.90000 3205.19000  28.9%  44.1 2544s
1002289 144253 3209.46000   54   19 4509.90000 3205.19000  28.9%  44.0 2548s
1003925 144461 3205.32286   52   36 4509.90000 3205.19000  28.9%  44.0 2553s
1005375 144597 3209.19000   63   14 4509.90000 3205.19000  28.9%  44.0 2557s
1006867 144816 3207.26000   46   16 4509.90000 3205.19000  28.9%  44.0 2562s
1008427 145022     cutoff   43       4509.90000 3205.19000  28.9%  44.0 2566s
1009897 145184 infeasible   41       4509.90000 3205.19000  28.9%  44.0 2571s
1011411 145385 3209.33000   64   14 4509.90000 3205.19000  28.9%  44.0 2575s
1012627 145667 3209.19000   53   16 4509.90000 3205.19000  28.9%  44.0 2580s
1016138 146170     cutoff   51       4509.90000 3205.19000  28.9%  44.0 2588s
1017750 146364     cutoff   68       4509.90000 3205.19000  28.9%  44.0 2594s
1019403 146536     cutoff   52       4509.90000 3205.19000  28.9%  44.0 2599s
1020924 146684 infeasible   60       4509.90000 3205.19000  28.9%  44.0 2604s
1022388 146844     cutoff   42       4509.90000 3205.19000  28.9%  44.0 2608s
1023759 147026 3207.34125   51   24 4509.90000 3205.19000  28.9%  43.9 2613s
1025226 147290 3207.26000   57   13 4509.90000 3205.19000  28.9%  43.9 2617s
```

1026690	147522	cutoff	51		4509.90000	3205.19000	28.9%	43.9	2622s
1028086	147806	cutoff	61		4509.90000	3205.19000	28.9%	43.9	2626s
1029498	147962	3209.47053	56	21	4509.90000	3205.19000	28.9%	43.9	2630s
1032267	148270	cutoff	55		4509.90000	3205.19000	28.9%	43.9	2639s
1033607	148471	3205.19000	52	22	4509.90000	3205.19000	28.9%	43.9	2643s
1035140	148580	cutoff	48		4509.90000	3205.19000	28.9%	43.9	2647s
1036581	148806	3207.26000	53	21	4509.90000	3205.19000	28.9%	43.9	2652s
1038169	148933	cutoff	56		4509.90000	3205.19000	28.9%	43.9	2658s
1039823	149063	cutoff	48		4509.90000	3205.19000	28.9%	43.9	2662s
1041121	149228	cutoff	70		4509.90000	3205.19000	28.9%	43.9	2666s
1042682	149380	cutoff	65		4509.90000	3205.19000	28.9%	43.9	2671s
1043960	149501	cutoff	54		4509.90000	3205.19000	28.9%	43.9	2675s
1045458	149723	infeasible	47		4509.90000	3205.19000	28.9%	43.9	2680s
1047111	149892	cutoff	68		4509.90000	3205.19000	28.9%	43.9	2685s
1050276	150436	infeasible	48		4509.90000	3205.19000	28.9%	43.9	2694s
1051923	150774	3208.50993	56	25	4509.90000	3205.19000	28.9%	43.9	2698s
1053580	150940	3209.19000	49	20	4509.90000	3205.19000	28.9%	43.8	2702s
1054728	151172	cutoff	71		4509.90000	3205.19000	28.9%	43.8	2707s
1056165	151306	3205.22285	48	37	4509.90000	3205.19000	28.9%	43.8	2711s
1057614	151582	3209.33000	53	20	4509.90000	3205.19000	28.9%	43.8	2717s
1059253	151773	cutoff	56		4509.90000	3205.19000	28.9%	43.8	2721s
1060757	151939	cutoff	59		4509.90000	3205.19000	28.9%	43.8	2725s
1062067	152119	cutoff	51		4509.90000	3205.19000	28.9%	43.8	2730s
1065075	152536	3205.28198	44	25	4509.90000	3205.19000	28.9%	43.8	2739s
1066573	152690	cutoff	55		4509.90000	3205.19000	28.9%	43.8	2743s
1067806	152882	cutoff	52		4509.90000	3205.19000	28.9%	43.8	2747s
1069227	153010	3206.19706	53	37	4509.90000	3205.19000	28.9%	43.8	2752s
1070657	153267	cutoff	48		4509.90000	3205.19000	28.9%	43.8	2756s
1072100	153486	cutoff	55		4509.90000	3205.19000	28.9%	43.8	2760s
1073659	153692	cutoff	59		4509.90000	3205.19000	28.9%	43.8	2765s
1076652	154007	3207.45611	48	19	4509.90000	3205.19000	28.9%	43.8	2773s
1077609	154239	3207.26000	51	19	4509.90000	3205.19000	28.9%	43.8	2779s
1079072	154467	3207.26000	53	25	4509.90000	3205.19000	28.9%	43.8	2783s
1080526	154691	3205.52031	51	34	4509.90000	3205.19000	28.9%	43.8	2787s
1081952	154923	cutoff	57		4509.90000	3205.19000	28.9%	43.8	2792s
1083405	155046	cutoff	66		4509.90000	3205.19000	28.9%	43.8	2796s
1084808	155310	3205.19000	49	22	4509.90000	3205.19000	28.9%	43.8	2801s
1086457	155428	cutoff	63		4509.90000	3205.19000	28.9%	43.8	2806s
1087892	155672	cutoff	48		4509.90000	3205.19000	28.9%	43.8	2810s
1089501	155803	3205.19000	47	24	4509.90000	3205.19000	28.9%	43.8	2815s
1090968	155922	3205.19000	48	29	4509.90000	3205.19000	28.9%	43.8	2820s
1094263	156290	cutoff	65		4509.90000	3205.19000	28.9%	43.8	2828s
1095567	156469	cutoff	47		4509.90000	3205.19000	28.9%	43.8	2833s
1097069	156691	cutoff	53		4509.90000	3205.19000	28.9%	43.8	2838s
1098599	156868	cutoff	69		4509.90000	3205.19000	28.9%	43.8	2843s
1099842	157066	3208.52571	71	29	4509.90000	3205.19000	28.9%	43.8	2847s
1101467	157293	cutoff	62		4509.90000	3205.19000	28.9%	43.8	2852s
1103040	157570	3209.33000	63	19	4509.90000	3205.19000	28.9%	43.8	2856s
1104555	157789	cutoff	50		4509.90000	3205.19000	28.9%	43.8	2861s
1106041	157911	3208.36000	57	18	4509.90000	3205.19000	28.9%	43.8	2866s
1107357	158137	3205.19000	53	21	4509.90000	3205.19000	28.9%	43.8	2870s
1108859	158331	cutoff	63		4509.90000	3205.19000	28.9%	43.8	2875s
1111782	158725	3205.62306	52	35	4509.90000	3205.19000	28.9%	43.8	2883s
1113257	159003	3207.37048	56	25	4509.90000	3205.19000	28.9%	43.8	2888s
1114793	159306	cutoff	60		4509.90000	3205.19000	28.9%	43.7	2892s
1116397	159443	3207.12000	48	29	4509.90000	3205.19000	28.9%	43.7	2898s
1117902	159606	3211.40000	60	16	4509.90000	3205.19000	28.9%	43.7	2903s
1119392	159678	3206.65492	45	23	4509.90000	3205.19000	28.9%	43.7	2908s
1120653	159885	3205.59333	55	28	4509.90000	3205.19000	28.9%	43.7	2912s

```
1122159 160100      cutoff   60      4509.90000 3205.19000   28.9%   43.7 2917s
1123646 160332 infeasible   50      4509.90000 3205.19000   28.9%   43.7 2921s
1125279 160444      cutoff   51      4509.90000 3205.19000   28.9%   43.7 2926s
1126783 160632 3205.19000   55   14 4509.90000 3205.19000   28.9%   43.7 2931s
1128435 160793      cutoff   54      4509.90000 3205.19000   28.9%   43.7 2935s
1130771 160952 3205.50543   46   31 4509.90000 3205.19000   28.9%   43.7 2940s
1132613 161260 3205.19000   48   30 4509.90000 3205.19000   28.9%   43.7 2948s
1134085 161465      cutoff   51      4509.90000 3205.19000   28.9%   43.7 2953s
1135610 161721 3205.19000   42   29 4509.90000 3205.19000   28.9%   43.7 2959s
1137138 161920 4174.26303   50   34 4509.90000 3205.19000   28.9%   43.7 2964s
1138537 162083 3207.14681   69   35 4509.90000 3205.19000   28.9%   43.7 2968s
1140016 162276 3205.19000   42   24 4509.90000 3205.19000   28.9%   43.7 2973s
1141519 162438      cutoff   52      4509.90000 3205.19000   28.9%   43.7 2977s
1142897 162611 3205.19000   47   23 4509.90000 3205.19000   28.9%   43.7 2982s
1144516 162770 3208.38000   62   17 4509.90000 3205.19000   28.9%   43.7 2987s
1145922 162925 3205.19000   44   27 4509.90000 3205.19000   28.9%   43.7 2991s
1147455 163082 3205.22314   47   38 4509.90000 3205.19000   28.9%   43.7 2996s
1148946 163159      cutoff   56      4509.90000 3205.19000   28.9%   43.7 3001s
1150390 163337      cutoff   62      4509.90000 3205.19000   28.9%   43.7 3005s
1151855 163503      cutoff   54      4509.90000 3205.19000   28.9%   43.7 3010s
1153365 163568 infeasible   59      4509.90000 3205.19000   28.9%   43.7 3015s
1156208 163973 3205.33350   60   39 4509.90000 3205.19000   28.9%   43.7 3025s
1159291 164347 3207.26000   60   21 4509.90000 3205.19000   28.9%   43.7 3033s
1160722 164498 3210.42000   52   14 4509.90000 3205.19000   28.9%   43.7 3037s
1162095 164657 3205.23221   52   30 4509.90000 3205.19000   28.9%   43.7 3042s
1163504 164882      cutoff   48      4509.90000 3205.19000   28.9%   43.7 3047s
1165080 165051 3207.40946   49   29 4509.90000 3205.19000   28.9%   43.7 3051s
1166549 165216 3205.31993   61   28 4509.90000 3205.19000   28.9%   43.7 3055s
1168005 165344      cutoff   63      4509.90000 3205.19000   28.9%   43.7 3060s
1170988 165616      cutoff   51      4509.90000 3205.19000   28.9%   43.7 3068s
1172392 165706      cutoff   50      4509.90000 3205.19000   28.9%   43.7 3073s
1173562 165905 3205.32558   59   23 4509.90000 3205.19000   28.9%   43.7 3077s
1174996 166210      cutoff   52      4509.90000 3205.19000   28.9%   43.7 3082s
1176681 166504 3207.26000   44   20 4509.90000 3205.19000   28.9%   43.7 3088s
1178272 166707      cutoff   67      4509.90000 3205.19000   28.9%   43.6 3092s
1179718 166823      cutoff   54      4509.90000 3205.19000   28.9%   43.6 3096s
1180977 167012      cutoff   66      4509.90000 3205.19000   28.9%   43.6 3101s
1182418 167106      cutoff   51      4509.90000 3205.19000   28.9%   43.6 3106s
1183904 167315 infeasible   44      4509.90000 3205.19000   28.9%   43.6 3110s
1185546 167510 3205.19000   47   23 4509.90000 3205.19000   28.9%   43.6 3115s
1188435 167959 3205.19000   60   21 4509.90000 3205.19000   28.9%   43.6 3124s
1190046 168166 3205.41898   48   33 4509.90000 3205.19000   28.9%   43.6 3128s
1191505 168410      cutoff   49      4509.90000 3205.19000   28.9%   43.6 3133s
1193085 168611      cutoff   54      4509.90000 3205.19000   28.9%   43.6 3137s
1194543 168784 infeasible   47      4509.90000 3205.19000   28.9%   43.6 3141s
1195952 168953 3205.41405   56   32 4509.90000 3205.19000   28.9%   43.6 3147s
1197543 169149 3205.19000   38   23 4509.90000 3205.19000   28.9%   43.6 3151s
1198934 169300      cutoff   65      4509.90000 3205.19000   28.9%   43.6 3155s
1200165 169415      cutoff   60      4509.90000 3205.19000   28.9%   43.6 3160s
1201663 169598 infeasible   49      4509.90000 3205.19000   28.9%   43.6 3165s
1204665 169953 3205.29843   60   20 4509.90000 3205.19000   28.9%   43.6 3173s
1206178 170194 3208.27693   55   26 4509.90000 3205.19000   28.9%   43.6 3178s
1207860 170321 3207.26000   46   18 4509.90000 3205.19000   28.9%   43.6 3182s
1209397 170488 3207.58623   61   35 4509.90000 3205.19000   28.9%   43.6 3187s
1210909 170600      cutoff   44      4509.90000 3205.19000   28.9%   43.6 3191s
1212037 170779 3205.39253   52   28 4509.90000 3205.19000   28.9%   43.6 3196s
1213445 170977      cutoff   52      4509.90000 3205.19000   28.9%   43.6 3200s
1216038 171211 3207.16451   68   35 4509.90000 3205.19000   28.9%   43.6 3211s
1217650 171360      cutoff   45      4509.90000 3205.19000   28.9%   43.6 3216s
```

1219037	171531	cutoff	57		4509.90000	3205.19000	28.9%	43.6	3221s
1220559	171579	3207.33747	63	26	4509.90000	3205.19000	28.9%	43.6	3227s
1221068	171687	cutoff	64		4509.90000	3205.19000	28.9%	43.6	3230s
1221908	171846	3209.38492	65	22	4509.90000	3205.19000	28.9%	43.6	3235s
1223292	172138	cutoff	75		4509.90000	3205.19000	28.9%	43.6	3240s
1225260	172322	3207.37304	61	22	4509.90000	3205.19000	28.9%	43.5	3246s
1227105	172512	cutoff	53		4509.90000	3205.19000	28.9%	43.5	3251s
1228862	172684	cutoff	57		4509.90000	3205.19000	28.9%	43.5	3256s
1230441	172811	3207.12000	50	18	4509.90000	3205.19000	28.9%	43.5	3261s
1231917	172997	cutoff	52		4509.90000	3205.19000	28.9%	43.5	3266s
1233622	173117	cutoff	41		4509.90000	3205.19000	28.9%	43.5	3271s
1235179	173235	cutoff	52		4509.90000	3205.19000	28.9%	43.5	3277s
1236531	173405	3207.26000	50	22	4509.90000	3205.19000	28.9%	43.5	3282s
1238164	173634	cutoff	50		4509.90000	3205.19000	28.9%	43.5	3287s
1239778	173849	infeasible	52		4509.90000	3205.19000	28.9%	43.5	3291s
1241163	174021	3205.19000	49	26	4509.90000	3205.19000	28.9%	43.5	3296s
1242693	174213	cutoff	60		4509.90000	3205.19000	28.9%	43.5	3300s
1244213	174384	infeasible	63		4509.90000	3205.19000	28.9%	43.5	3305s
1245796	174534	3207.26000	55	13	4509.90000	3205.19000	28.9%	43.5	3310s
1247284	174750	3205.29372	41	36	4509.90000	3205.19000	28.9%	43.5	3315s
1250183	174984	3207.26000	47	20	4509.90000	3205.19000	28.9%	43.5	3320s
1252032	175378	cutoff	55		4509.90000	3205.19000	28.9%	43.5	3329s
1253530	175507	3207.26000	55	23	4509.90000	3205.19000	28.9%	43.5	3333s
1255046	175678	3209.19000	51	17	4509.90000	3205.19000	28.9%	43.5	3339s
1256600	175816	cutoff	52		4509.90000	3205.19000	28.9%	43.5	3343s
1258050	175952	3207.66020	54	21	4509.90000	3205.19000	28.9%	43.5	3347s
1259285	176089	cutoff	47		4509.90000	3205.19000	28.9%	43.5	3352s
1260768	176238	cutoff	56		4509.90000	3205.19000	28.9%	43.5	3356s
1262277	176401	cutoff	55		4509.90000	3205.19000	28.9%	43.5	3361s
1263825	176521	3207.91057	66	20	4509.90000	3205.19000	28.9%	43.5	3366s
1265174	176739	3205.21663	43	31	4509.90000	3205.19000	28.9%	43.5	3371s
1266799	176883	cutoff	51		4509.90000	3205.19000	28.9%	43.5	3375s
1268344	177103	3207.52664	56	26	4509.90000	3205.19000	28.9%	43.5	3380s
1271277	177431	cutoff	49		4509.90000	3205.19000	28.9%	43.5	3388s
1272828	177629	3205.19000	56	23	4509.90000	3205.19000	28.9%	43.5	3392s
1274326	177804	3205.19000	55	19	4509.90000	3205.19000	28.9%	43.5	3398s
1275913	177970	cutoff	52		4509.90000	3205.19000	28.9%	43.5	3403s
1277231	178201	cutoff	50		4509.90000	3205.19000	28.9%	43.5	3407s
1278829	178319	infeasible	60		4509.90000	3205.19000	28.9%	43.5	3412s
1280150	178387	3207.12000	71	24	4509.90000	3205.19000	28.9%	43.5	3416s
1281620	178564	cutoff	52		4509.90000	3205.19000	28.9%	43.5	3421s
1283305	178756	3209.33000	59	19	4509.90000	3205.19000	28.9%	43.5	3425s
1286246	179231	3208.33300	55	18	4509.90000	3205.19000	28.9%	43.5	3433s
1287840	179473	3209.33000	60	16	4509.90000	3205.19000	28.9%	43.5	3438s
1289360	179676	cutoff	60		4509.90000	3205.19000	28.9%	43.5	3442s
1290908	179819	3207.32745	55	25	4509.90000	3205.19000	28.9%	43.5	3446s
1292426	179978	cutoff	51		4509.90000	3205.19000	28.9%	43.5	3451s
1293711	180159	cutoff	63		4509.90000	3205.19000	28.9%	43.5	3456s
1295161	180309	3207.34870	50	20	4509.90000	3205.19000	28.9%	43.5	3461s
1296691	180493	cutoff	52		4509.90000	3205.19000	28.9%	43.5	3466s
1298246	180598	3209.68000	61	15	4509.90000	3205.19000	28.9%	43.5	3470s
1300996	180979	cutoff	52		4509.90000	3205.19000	28.9%	43.5	3479s
1302629	181397	cutoff	61		4509.90000	3205.19000	28.9%	43.4	3484s
1304398	181788	3207.32154	61	23	4509.90000	3205.19000	28.9%	43.4	3488s
1306084	182093	cutoff	51		4509.90000	3205.19000	28.9%	43.4	3493s
1307743	182281	3207.26000	53	32	4509.90000	3205.19000	28.9%	43.4	3497s
1308973	182525	cutoff	56		4509.90000	3205.19000	28.9%	43.4	3501s
1310463	182820	3210.20093	57	21	4509.90000	3205.19000	28.9%	43.4	3506s
1312070	183051	cutoff	44		4509.90000	3205.19000	28.9%	43.4	3510s

```
 1313612 183212 3205.19000   44   20 4509.90000 3205.19000  28.9%  43.4 3516s
 1315034 183478     cutoff   59      4509.90000 3205.19000  28.9%  43.4 3521s
 1316724 183696     cutoff   51      4509.90000 3205.19000  28.9%  43.4 3525s
 1318191 183950 3205.19000   35   34 4509.90000 3205.19000  28.9%  43.4 3530s
 1320367 184194     cutoff   60      4509.90000 3205.19000  28.9%  43.4 3535s
 1322876 184525     cutoff   46      4509.90000 3205.19000  28.9%  43.4 3543s
 1323996 184673 3209.38838   62   21 4509.90000 3205.19000  28.9%  43.4 3547s
 1325254 184892 4359.85853   52   46 4509.90000 3205.19000  28.9%  43.4 3551s
 1326693 185118 3209.33000   62   15 4509.90000 3205.19000  28.9%  43.4 3555s
 1328184 185277 3207.65108   60   27 4509.90000 3205.19000  28.9%  43.4 3560s
 1329623 185477 3205.19000   55   24 4509.90000 3205.19000  28.9%  43.4 3565s
 1332552 185796 3206.44538   49   34 4509.90000 3205.19000  28.9%  43.4 3573s
 1333954 185923 3207.36278   66   21 4509.90000 3205.19000  28.9%  43.4 3579s
 1335355 186056 3207.26000   54   28 4509.90000 3205.19000  28.9%  43.4 3584s
 1336897 186201     cutoff   57      4509.90000 3205.19000  28.9%  43.4 3589s
 1338261 186336 3205.19000   54   26 4509.90000 3205.19000  28.9%  43.4 3593s
 1339828 186481 3207.38664   56   24 4509.90000 3205.19000  28.9%  43.4 3598s
 1341333 186578     cutoff   68      4509.90000 3205.19000  28.9%  43.4 3600s

Cutting planes:
  Learned: 11
  Gomory: 92
  Lift-and-project: 230
  Cover: 910
  Implied bound: 60
  Projected implied bound: 6
  Clique: 43
  MIR: 529
  Mixing: 56
  StrongCG: 3
  Flow cover: 701
  GUB cover: 28
  Inf proof: 27
  Zero half: 20
  Mod-K: 1
  RLT: 35
  Relax-and-lift: 25
  BQP: 1

Explored 1342002 nodes (58202890 simplex iterations) in 3600.35 seconds (1569.03 work units)
Thread count was 20 (of 20 available processors)

Solution count 10: 4509.9 4553.78 4553.78 ... 5909.28

Time limit reached
Best objective 4.509900000000e+03, best bound 3.205190000000e+03, gap 28.9299%
```