Extended Capacitated Warehouse Location

Carolina Dias[1], Mariana Pereira[1] and Sofia Fernandes[1]

[1] Faculty of Engineering, University of Porto, Portugal

Abstract. Tackling complex combinatorial problems requires selecting effective solution methodologies. This work uses the Capacitated Warehouse Location Problem (CWLP) as a benchmark to compare Mixed-Integer Programming (MIP) and Constraint Programming (CP). An extended formulation is proposed, introducing soft penalization constraints on warehouse co-openings and regional interactions. Computational experiments on OR-Library benchmark instances of increasing size evaluate both approaches in terms of runtime, solution quality and solver status. The results show that MIP outperforms CP in terms of scalability, robustness and solution quality, regardless of instance size.

Keywords: Capacitated Warehouse Location, Mixed-Integer Programming, Constraint Programming.

1 Introduction

The Capacitated Warehouse Location Problem (CWLP) is a fundamental combinatorial optimization challenge in operations research and supply chain management. In its standard form, the problem aims to determine the optimal set of facilities to open from a discrete set of possible locations, as well as the assignment of client demands to these open facilities. The primary objective is to minimize the total cost, which comprises the fixed costs associated with opening warehouses and the transportation costs required to serve customer demand, subject to strict capacity constraints at each facility [1].

However, practical logistical decisions rarely depend solely on direct costs and capacity limits. Real-world facility location often entails complex structural considerations, such as regional dispersion and accessibility, and strategic redundancy. For instance, an organization may wish to discourage the opening of warehouses that are too close to one another to avoid market cannibalization, or in turn, manage the intensity of operations across different administrative regions to ensure balanced service levels.

This report aims to address these complexities by exploring two distinct modeling paradigms: Mixed-Integer Programming (MIP) and Constraint Programming (CP). We first benchmark these approaches on the standard CWLP to establish a performance baseline. Subsequently, we evaluate their efficacy on an extended CWLP, which incorporates realistic operational constraints. Specifically, this extension introduces soft incompatibility penalties between warehouse pairs and regional intensity penalties to regulate the distribution of facilities across territories. The study analyzes how these additional constraints influence the solver's performance and solution quality.

2 The MIP Model

This section presents the mathematical formulations used to solve the warehouse location problem. We begin with the standard linear model and subsequently introduce the extended formulation that incorporates logical and regional constraints.

2.1 Original MIP Formulation

The standard CWLP is modeled as a MIP problem. Let $I = \{1, \dots, m\}$ be the set of potential warehouse locations and $J = \{1, \dots, n\}$ be the set of customers.

The model parameters are defined as follows:

F_i: The fixed cost incurred to open warehouse $i \in I$.
Q_i: The maximum capacity of warehouse $i \in I$.
d_j: The demand associated with customer $j \in J$
c_{ij}: The allocation cost to serve the full demand of customer j from warehouse i.

Decision Variables
The model employs two sets of binary decision variables to represent the strategic and operational choices:

$y_i \in \{0, 1\}$: A binary variable that indicates the activation status of warehouse i. It takes value 1 if warehouse i is opened, and 0 otherwise.
$x_{ij} \in \{0, 1\}$: A binary variable representing the allocation of customer demand. It takes value 1 if customer j is assigned to warehouse i, and 0 otherwise.

Mathematical Model
The objective is to minimize the total system cost, which is the sum of fixed facility costs and variable allocation costs:

$$\min Z = \sum_{i \in I} F_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \tag{1}$$

Subject to the following constraints:

Demand Satisfaction: ensures that each client is allocated to exactly one warehouse.

$$\sum_{i \in I} x_{ij} = 1, \qquad \forall j \in J \tag{2}$$

Capacity Constraints: ensures that the total demand assigned to warehouse i does not exceed its capacity Q_iFurthermore, it prevents any demand from being assigned to a closed warehouse ($y_i = 0$).

$$\sum_{j \in J} d_j x_{ij} \leq Q_i y_i, \qquad \forall i \in I \tag{3}$$

Linking Constraints: reinforces the logical requirement that a customer j can only be serviced by warehouse i if that warehouse is currently active.

$$x_{ij} \leq y_i, \qquad \forall i \in I, \forall j \in J \tag{4}$$

Variable Domains:

$$y_i \in \{0,1\}, x_{ij} \in \{0,1\}, \qquad \forall i \in I, \forall j \in J \tag{5}$$

2.2 Extended MIP Formulation

To address the practical logistical constraints outlined in the introduction, the original formulation is augmented with additional structural requirements. This extended model incorporates soft incompatibility penalties between specific warehouse pairs and regional intensity constraints to regulate facility distribution.

Model Extensions

The extended model introduces the following sets and parameters:

$S \subseteq I \times I$: A set of warehouse pairs (i,k) with $i < k$ that are subject to soft incompatibility penalties (e.g., due to proximity or sharing the same resources).

P_{ik}: The penalty cost incurred if both warehouses i and k are opened simultaneously, for $(i,k) \in S$.

$R = \{1, \dots, r\}$: A set of geographical regions. Let $I_r \subseteq I$ be the subset of warehouses located in region r. The number of regions r depends on the instance size - $r = 6$ for $(m = 16 \vee m = 25)$, $r = 9$ for $m = 50$, and $r = 12$ for $m = 100$.

$C \subseteq I \times I$: A set of cross-regional warehouse pairs subject to intensity penalties.

$P_{reg(i,k)}$: The penalty cost associated with operating warehouses i and k simultaneously if they belong to a penalized region pair in C.

New Variables

In addition to the standard variables y_i and x_{ij}, the model introduces auxiliary variables to linearize the logical conditions associated with simultaneous facility openings:

$z_{ik} \in \{0,1\}$: A binary variable equal to 1 if both warehouses i and k are opened, and 0 otherwise, for all $(i,k) \in S$.

$\delta_{ik} \in \{0,1\}$: A binary variable equal to 1 if both warehouses i and k are opened, and 0 otherwise, for all $(i,k) \in C$.

Mathematical Model

The objective function is extended to minimize the sum of the original costs plus the newly introduced penalty terms:

$$\min Z_{ext} = \mathrm{Z} + \sum_{(i,k) \in S} P_{ik} z_{ik} + \sum_{(i,k) \in C} P_{reg(i,k)} \delta_{ik} \tag{6}$$

This objective is subject to the original constraints (2)-(5) defined in Section 2.1, along with the following additional constraints:

Soft Incompatibility Linearization: To ensure the logical condition $z_{ik} = 1 \Leftrightarrow (y_i = 1 \wedge y_k = 1)$, the following linear constraints are imposed for all $(i,k) \in S$:

$$z_{ik} \geq y_i + y_k - 1 \tag{7}$$

$$z_{ik} \leq y_i \tag{8}$$

$$z_{ik} \leq y_k \tag{9}$$

Regional Coverage Constraints: To ensure minimum service levels across all territories, the model enforces that at least one warehouse is opened in each region:

$$\sum_{i \in I_r} y_i \geq 1, \qquad \forall \mathrm{r} \in \mathrm{R} \tag{10}$$

Cross-Regional Intensity Linearization: Similar to the soft incompatibilities, the cross-regional penalty variables δ_{ik} are linked to the activation variables y for all $(i,k) \in C$:

$$\delta_{ik} \geq y_i + y_k - 1 \tag{11}$$

$$\delta_{ik} \leq y_i \tag{12}$$

$$\delta_{ik} \leq y_k \tag{13}$$

Variable Domains:

$$z_{ik} \in \{0,1\}, \qquad \delta_{ik} \in \{0,1\} \tag{14}$$

3 The CP Model

The CP model differs fundamentally from the MIP model in terms of conceptualization and solution strategy. While the MIP model relies on linear objective functions and constraints, and optimizes the objective function using relaxations refined by branch-and-bound, the CP model enforces feasibility through discrete constraints and explores the solution space using constraint propagation and backtracking [2].

In the problem considered in this work, the MIP formulation is based on binary decision variables, with capacity and linking constraints being enforced through inequalities. Feasibility and optimality are driven by the strength of the linear relaxation [3].

By contrast, the CP formulation relies on Boolean variables and enforces feasibility directly using logical and combinatorial constraints. The CWLP constraints are expressed as feasibility conditions rather than inequalities, allowing infeasible solutions (e.g., exceeding warehouse capacity, customers assigned to closed facilities) to be pruned early in the search process through constraint propagation. Thus, the CP model explores the solution space through discrete logic aligned with the structure of the instances studied in this work.

3.1 Original CP Formulation

The CP model addresses the same CWLP introduced in the MIP formulation and relies on the same set, parameters, decision variables, and objective function defined in Section 2. The variables $y_i$and x_{ij} preserve their interpretation as facility activation and client assignment decisions, respectively, and the goal remains the cost minimization.

The constraints of the CP model are logically equivalent to those defined in the MIP formulation. Respectively, demand satisfaction, capacity and linking constraints are imposed through exclusivity constraints, domain-based feasibility relations, and logical implications between variables.

3.2 Extended CP Formulation

The extended CWLP formulation introduces constraints reflecting practical considerations outlined in Section 1, such as regional assignment rules and intensity limits. As in the extended MIP formulation, these constraints further restrict the set of feasible solutions without altering the core decision variables. Within the CP framework, they are computed as logical conditions over the variable domains. These added restrictions increase the density and tightness of the constraint network while avoiding auxiliary variables and large constant coefficients usually required in linear formulations.

4 An analysis of the tests and results

This chapter presents a comprehensive analysis of the computational experiments conducted to evaluate the performance of the proposed models and solution approaches. The objective of these tests is to assess how the original and extended formulations behave under increasing dimensions and to compare the efficiency and robustness of Mixed-Integer Programming (MIP) and Constraint Programming (CP) approaches.

Particular attention is given to scalability, the impact of the extended formulation on computational effort and solution quality, and the role of different CP search strategies in solving instances of varying size and complexity. The analysis is supported by runtime plots and comparative tables derived from the experimental results, enabling a structured interpretation of observed performance trends.

4.1 Experimental setup and evaluation criteria

This section describes the experimental framework adopted to evaluate the performance of the proposed optimization models. The analysis considers both the original CWLP and its extended formulation. All computational experiments are designed to ensure consistency and comparability across models, solvers, and instance sizes.

The experimental study uses benchmark instances derived from the OR-Library dataset [4]. For each problem dimension, two distinct instances are selected, allowing the results to reflect scalability with respect to size and variability across instances with similar dimensions. The dimensions include small-scale instances with 16 warehouses and 50 customers, medium-scale instances with 25 and 50 warehouses serving 50 customers, and a large-scale instance with 100 warehouses and 1000 customers. This selection enables assessment of solver behavior under increasing combinatorial complexity while avoiding conclusions based on a single instance per size.

Two modeling paradigms are evaluated: MIP, solved using IBM ILOG CPLEX, and CP, solved using IBM CP Optimizer. For the CP approach, three search strategies are tested - Depth First, Restart, and MultiPoint - to analyze robustness and scalability under added complexity. All experiments run under identical conditions, using a single processing thread and a uniform time limit of 400 s. When the time limit is reached, the best feasible solution found by the solver is recorded.

The evaluation of results is based on a set of Key Performance Indicators (KPIs) capturing computational efficiency and solution quality. These include solver runtime, objective value of the best solution found, solver termination status, and, when applicable, the relative difference between solutions obtained by different approaches. Structural indicators introduced in the extended formulation are also considered. These KPIs are reported through comparative tables and plots, providing a comprehensive basis for the analysis presented in the following sections.

4.2 Performance analysis of the original formulation

The computational performance of the original formulation is analyzed using per-instance results reported in Table 1 in the Appendix. This table summarizes the KPIs obtained for each benchmark instance, including runtime, objective value, and solver termination status for the MIP model and the best-performing CP strategy.

The results show that the MIP model consistently solves all small- and medium-scale instances to proven optimality within short computational times. This behavior remains stable across all instances of each dimension (16×50, 25×50, and 50×50), indicating strong robustness with respect to instance-specific features. For the largest instances (100×1000), the MIP solver reaches the time limit and returns high-quality feasible solutions, reflecting the sharp increase in combinatorial complexity at that scale.

In contrast, CP approaches struggle to scale even for the original formulation. For most instances, CP reaches the time limit and returns feasible but suboptimal solutions. Although different CP search strategies emerge as the best-performing option depending on the instance, the overall performance gap with respect to MIP remains significant as problem size increases.

Overall, the results reported in Table 1 establish MIP as the more robust and scalable paradigm for the original formulation, providing a solid baseline for evaluating the impact of the extended model introduced in the following section.

4.3 Impact of the extended formulation on computational performance

The impact of the CWLP extension on solver performance is evaluated by comparison with the original problem results. The extended model introduces soft penalization constraints on warehouse co-openings, increasing the number of decision variables and constraints and, consequently, the overall structural complexity of the problem.

The effect of this extension on the MIP approach is illustrated in Figure 1, which compares the runtime of the original and extended formulations as a function of problem dimension. For small- and medium-scale instances (16×50, 25×50, and 50×50), both formulations are solved efficiently; however, the extended formulation consistently requires slightly longer computational times, reflecting the increased structural and combinatorial complexity introduced by the additional constraints.

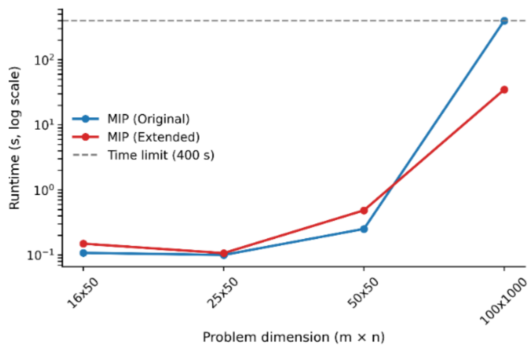

Fig. 1. Runtime comparison between the original and extended MIP formulations as a function of problem dimension

For the largest instances, a notable behavior is observed. While the original MIP formulation reaches the imposed time limit and returns high-quality feasible solutions, the extended formulation consistently attains optimality within the same computational budget.

In contrast, the impact of the extended formulation on the CP approach is markedly different. As reported in Table 2 in the Appendix, CP struggles to scale under the extended model. For medium and large instances, all CP search strategies frequently reach the time limit, often returning only feasible solutions, and in some cases failing to find any feasible solution at all. In particular, the Depth First strategy proves unsuitable for large extended instances, while Restart and MultiPoint strategies show greater robustness but still fail to achieve competitive performance when compared to MIP.

4.4 Discussion on solver performance and model suitability

The results show a clear performance advantage of the MIP approach over CP across all scenarios. Although CP relies on constraint propagation to prune the search space, this mechanism is not fully exploited due to the problem structure. The CWLP is dominated by linear assignment, capacity, and activation constraints that yield strong linear relaxations in the MIP framework, enabling effective pruning through branch-and-bound. In contrast, the corresponding CP constraints offer limited domain reduction - since most variables are binary, propagation cannot substantially reduce their domains, forcing the solver to rely heavily on search as problem size increases.

The extended formulation further reinforces this distinction. While soft penalization constraints increase model complexity, they also restrict the feasible region and reduce symmetry, improving convergence in the MIP model. In the CP model, however, these constraints mainly increase constraint density without strengthening propagation, leading to the exploration of many partial assignments and degraded scalability. Overall, these results indicate that solver choice should be guided by the dominant constraint structure - MIP is suitable for problems with linear cost and capacity-driven feasibility, whereas CP is more effective for problems with richer combinatorial or logical constraints that are difficult to capture in linear form.

5 Conclusions

This work analyzed the CWLP using MIP and CP, considering both a standard formulation and an extended variant with soft penalization of warehouse co-openings and regional interactions. The objective was to evaluate the scalability and robustness of both approaches under different levels of model complexity.

The computational results show that the MIP approach consistently outperforms CP across all tested instances. This behavior is primarily due to the linear structure of the (original and extended) CWLP, which allows MIP solvers to exploit strong linear relaxations and efficiently prune the search space. In contrast, the CP model displayed limited domain reduction, resulting in extensive search and reduced scalability.

Overall, the results highlight the importance of aligning the modeling paradigm with the dominant problem structure. For CWLP variants characterized by linear costs and capacity-driven feasibility, MIP emerges as the most suitable approach. Future research may explore alternative extensions to better exploit CP, such as incorporating complex combinatorial constraints like mutually exclusive warehouse clusters or conditional activation rules, which can be efficiently handled through constraint propagation.

References

1. Beasley, J.E.: An algorithm for solving large capacitated warehouse location problems. Eur. J. Oper. Res. 33, 314–325 (1988). https://doi.org/10.1016/0377-2217(88)90175-0
2. Apt, K.R.: Principles of Constraint Programming. Cambridge University Press, Cambridge (2003)
3. Wah, B.W.: Mixed Integer Programming. In: Wiley Encyclopedia of Computer Science and Engineering. Wiley (2007). https://doi.org/10.1002/9780470050118.ecse244
4. OR-Library: http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/, last accessed 2026/01/13

Appendix

Table 1. Per-instance computational KPIs for the original formulation (MIP vs best CP strategy)

Instance	Dimension	MIP Obj	MIP Time (s)	MIP Status	Best CP Search	CP Obj	CP Time (s)	CP Status
cap61.txt	16x50	932,615.75	0.17	opt. sol.	Restart	932,615.75	40.71	Optimal
cap62.txt	16x50	977,799.40	0.04	opt. sol.	MultiPoint	977,799.40	400.16	Feasible
cap101.txt	25x50	796,648.44	0.12	opt. sol.	Restart	796,648.44	400.21	Feasible
cap102.txt	25x50	854,704.20	0.08	opt. sol.	Restart	854,704.20	400.13	Feasible
cap121.txt	50x50	793,439.56	0.21	opt. sol.	Restart	793,439.56	400.24	Feasible
cap122.txt	50x50	854,900.45	0.29	opt. sol.	Restart	858,235.89	400.26	Feasible
capa.txt_10000	100x1000	18,440,271.22	400.28	time-limited/feasible	Restart	34,899,342.19	403.33	Feasible
capa.txt_12000	100x1000	17,765,201.95	400.21	time-limited/feasible	Restart	27,995,809.35	405.48	Feasible

Table 2. Per-instance computational KPIs for the extended formulation (MIP vs best CP strategy)

Instance	Dimension	MIP Obj	MIP Time (s)	MIP Status	Best CP Search	CP Obj	CP Time (s)	CP Status
cap61.txt	16x50	943,376.30	0.18	opt. sol.	Restart	943,376.30	15.60	Optimal
cap62.txt	16x50	988,599.40	0.12	opt. sol.	Multipoint	988,599.40	400.06	Feasible
cap101.txt	25x50	822,819.61	0.13	opt. sol.	Restart	822,819.61	400.06	Feasible
cap102.txt	25x50	872,203.24	0.08	opt. sol.	Restart	872,203.24	400.07	Feasible
cap121.txt	50x50	818,542.64	0.18	opt. sol.	Restart	818,542.64	400.19	Feasible
cap122.txt	50x50	877,410.38	0.79	opt. sol.	Restart	877,410.38	400.08	Feasible
capa.txt_10000	100x1000	25,461,030.54	36.10	opt. sol.	MultiPoint	68,135,520.29	403.10	Feasible
capa.txt_12000	100x1000	25,461,030.54	33.63	opt. sol.	MultiPoint	66,845,704.83	407.08	Feasible