Supplementary Materials

S1 CQNs for System-directed and Manual Picking

This section presents the CQNs for system-directed policy and manual picking.

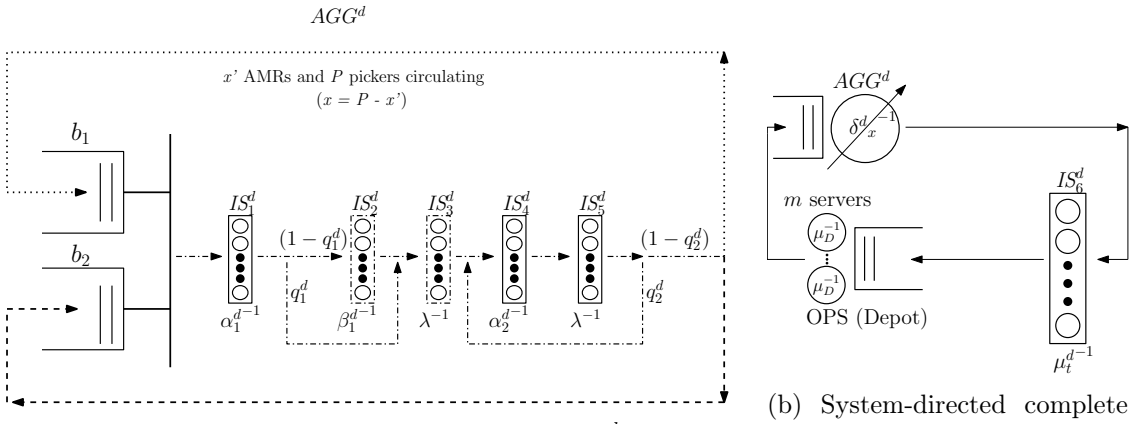
S1.1 System-directed

In the system-directed (with no zoning, NZ) policy, incoming orders are assigned to available AMRs, which then travel through the warehouse to collect the required items. Each AMR is paired with a picker, and together they complete all picks for the assigned order without restriction to specific warehouse zones. The picker and AMR move jointly through the pick locations. After completing the order, the AMR proceeds to the depot for unloading, and the picker can be paired with a new available AMR. This approach minimizes the number of human-robot matchings per order, simplifying collaboration, though it may lead to longer picker travel distances compared to the swarm policy.

Similar to the swarm policy in Section 3.2, the system-directed policy (with NZ) can also be represented as a CQN. Figure 25 illustrates the aggregation procedure for the network of the system-directed policy. Figure 25a shows that, once a picker is matched with an AMR, the picker begins traveling to the first pick location, modeled as IS_1^d, with an average travel time of $\alpha_1^{d^{-1}}$. There is a probability of q_1^d that the AMR arrives at the pick location before the picker. Conversely, with a probability of $(1 - q_1^d)$, the picker must wait for the AMR. This waiting time is captured in IS_2^d, with an average duration of $\beta_1^{d^{-1}}$.

Once both the AMR and the picker have arrived at the first pick location, the first item is picked, a process represented by IS_3^d with an average duration of λ^{-1}. After this, the AMR and picker proceed to subsequent pick locations. The travel time between each pick location is modeled as IS_4^d, with an average duration of $\alpha_2^{d^{-1}}$, while the picking process at each location is captured in IS_5^d. After each pick, with a probability of q_2^d, the order remains incomplete, requiring the retrieval of another item. Conversely, with a probability of $(1 - q_2^d)$, the order is considered complete.

Figure 25b shows the aggregated network of the system-directed policy. Once an order is fully picked, the AMR travels to the depot, a process represented by IS_6^d, with an average travel time of $\mu_t^{d^{-1}}$. Finally, the items are unloaded at the depot.



(a) System-directed picking network (AGG^d)

(b) System-directed complete network

Figure 25: Aggregation of the system-directed queuing network into AGG_d and the complete network

Similar to the Swarm-picking network, the throughput of System-directed picking network, δ_x^d , can be found using Little's law as follows:

$$\delta_x^d = \frac{l'\alpha_1^d\beta_1^d\lambda^2\alpha_2^d}{\beta_1^d\lambda^2\alpha_2^d + (1-q_1^d)\alpha_1^d\lambda^2\alpha_2^d + \alpha_1^d\beta_1^d\lambda\alpha_2^d + \frac{\alpha_1^d\beta_1^d\lambda^2}{1-q_2^d} + \frac{\alpha_1^d\beta_1^d\lambda\alpha_2^d}{1-q_2^d}} \tag{11}$$

Now, by replacing $\delta(x)$ and u by δ_x^d and 1, respectively in Equation 8, the steady states of System-directed complete network can be found as shown in Equation 12.

$$\Pi_{\gamma=(x',y,z)} = \frac{Z_\gamma}{\sum\limits_{\gamma\in\Gamma} Z_\gamma}, \text{ where } Z_\gamma = \frac{\prod_{n=1}^{R+1} a_n}{\prod_{n=R-y+1}^{R+1} a_n} \times \frac{\mu_D^{R-y} \times \prod_{j=x'}^{R-1} \delta_{P-j-1}^d \times {\mu_t^d}^{R-z} \times R!}{z!},$$

$$\text{and } a_n = \begin{cases} \min(h, R-n+1) & n = 1, 2, \ldots, R, \\ 1 & n = R+1 \end{cases} \tag{12}$$

Where $\gamma = (x', y, z)$ (# AMRs at AGG^d, # AMRs at depot, # AMRs at IS_6), and $x'+y+z = R$.

To validate the analytical results of system-directed policy, order throughputs from the system-directed policy model (Equation 12) are compared against a detailed real-world warehouse discrete event simulation (with real deterministic travel times between points based on distance) across 3000 problem instances. This simulation replicates the real-world operations of a warehouse, where pickers and AMRs move between various pick locations. Travel times are determined based on the actual distances between each pair of locations, rather than simulating the queuing model itself. The settings for these instances are summarized in Table 7. The absolute error percentage is computed as $(|TH_a - TH_s|/TH_s) \times 100$, where TH_a and TH_s denote

the analytical and simulated hourly throughputs, respectively. A summary of the error terms is provided in Table 8. The results show minimal discrepancy, with an average error of less than 0.33% for both order throughput (TH) and expected order throughput time (TPT).

Table 7: Settings of 3000 problem instances for validating system-directed analytical results

Parameter	# of settings	Details
Warehouse size	15	2, 4, 6, 8, and 10 aisles with 1, 2, and 3 blocks where $N = 30$
Item allocation	1	Random
Number of pickers	4	2, 4, 6, and 8 pickers
R/P ratio	5	1, 1.5, 2, 2.5 and 3
Velocities	2	Speed ratios of **1** ($v_r = 1$, $v_p = 1$) and 2 ($v_r = 1.33$, $v_p = 0.67$)
Order size (# SKUs)	5	2, 4, 6, 8, 10

Table 8: Absolute error percentages in throughput and throughput time of system-directed policy

Statistics	3000 instances (entire population)
TH average absolute error (%)	0.32 (95%CI = $[0.31, 0.34]$)
TH maximum error (%)	4.78
TPT average absolute error (%)	0.32
TPT maximum error (%)	5.02

S1.2 Manual Picking

Figure 26 illustrates the CQN for the manual picking process. In this model, we assume that there are sufficient unloading stations at the depot for manual pickers. This assumption is realistic, as manual pickers typically drop off loaded carts at the depot and immediately retrieve new carts, preventing the formation of queues.

In this network, the different components represent various time intervals:

IS_1^m models the travel time for pickers to reach the first pick location. IS_2^m captures the time required to pick the first item. IS_3^m models the travel time to the subsequent pick location. IS_4^m represents the picking time for that item. IS_5^m models the travel time to the depot, and IS_6^m captures the unloading time. Furthermore, if the order size exceeds one, additional picks are necessary. In such cases, we set $q_1^m = 1$ (and $q_1^m = 0$ when the order size is exactly one). When the order size O equals 1, we also assign $\alpha_2^m = 1$ and $\lambda_2 = 1$. For orders with $O > 1$, there is a possibility that the order remains incomplete after the second pick, necessitating further picks. To account for this possibility, we define: $q_2^m = (1 - \frac{1}{O-1})$.

The throughput of the manual picking network, denoted as δ_P^m, depends on the number of pickers and can be determined using Little's Law, as shown below:

$$\delta_P^m = \frac{P}{\mu_t^{m-1} + \mu_D^{-1} + \alpha_1^{m-1} + \lambda_1^{-1} + \frac{q_1^m}{1-q_2^m}\alpha_2^{m-1} + \frac{q_1^m}{1-q_2^m}\lambda_2^{-1}} \tag{13}$$

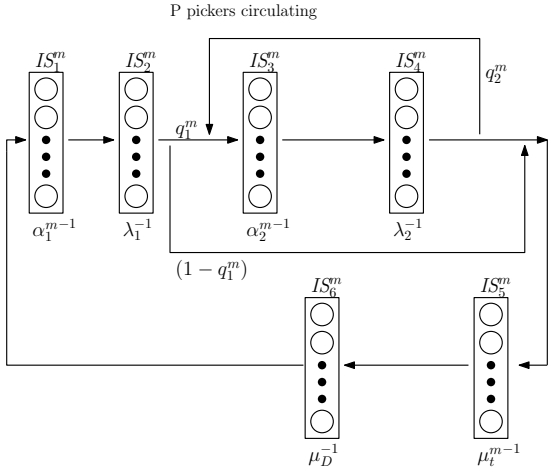


Figure 26: CQN representation of manual picking operations

To validate the manual picking analytical results, order throughputs from the manual picking model (Equation 13) are compared against a detailed discrete event simulation across 1440 problem instances. The simulation model is again based on the real warehouse, not the queuing network. It is similar to the model used for the swarm policy which captures the real warehouse operations and travel times. The settings for these instances are summarized in Table 9. The absolute error percentage is computed as $(|TH_a - TH_s|/TH_s) \times 100$, where TH_a and TH_s denote the analytical and simulated hourly throughputs, respectively. A summary of the error terms is provided in Table 10. The results show minimal discrepancy, with an average error of 0.16% for both order throughput (TH) and expected order throughput time (TPT).

Table 9: Settings of 1440 problem instances for validating manual picking analytical results

Parameter	# of settings	Details
Warehouse size	18	2, 4, 6, 8, 10, and 12 aisles with 1, 2, and 3 blocks where $N = 30$
Item allocation	1	Random
Number of pickers	8	2, 4, 6, 8, 10, 12, 14, and 16 pickers
Velocities	2	Speed of $v_p = 0.67$ and $v_p = 1$
Order size (# SKUs)	5	2, 4, 6, 8, 10

Table 10: Absolute error percentages in throughput and throughput time of manual picking policy

Statistics	1440 instances (entire population)
TH average absolute error (%)	0.16
TH maximum error (%)	0.83
TPT average absolute error (%)	0.16
TPT maximum error (%)	0.83

S2 Supporting Data and Illustrations

This appendix provides supporting materials including several figures, a table, and an example for this paper.

S2.1 Problem description figures

Here, the additional materials describing the problem are presented. Figure 27 shows the flowchart of the order picking under the swarm policy in which c and p are the number of AMRs and pickers in their buffer, respectively.

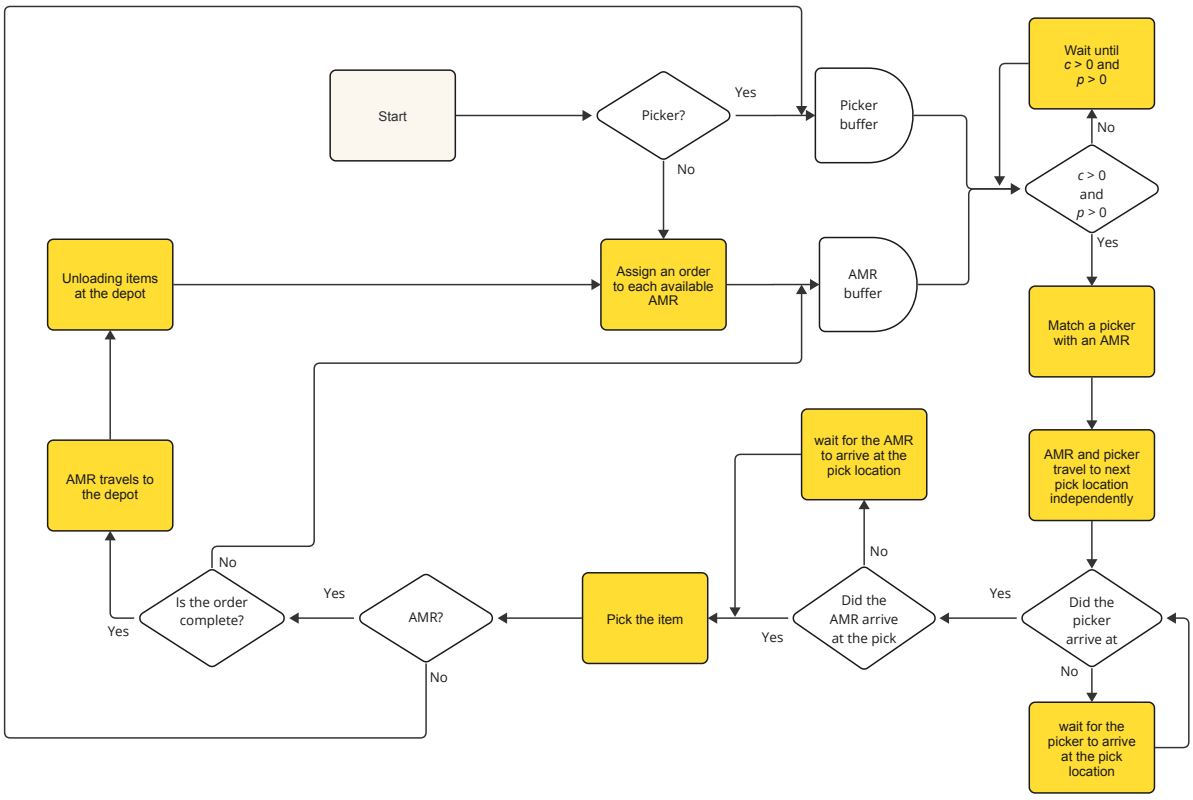
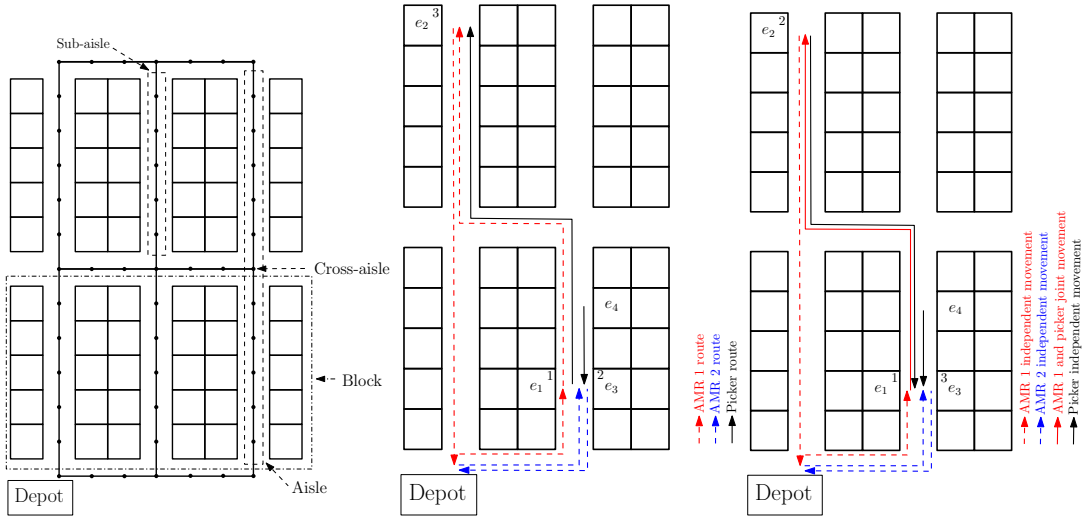


Figure 27: Flowchart of swarm policy operations from order assignment to unloading

Example 1 compares the makespan of fulfilling identical orders using three different picking policies: swarm, system-directed, and manual.

Example 1. Picking the same orders with different policies: Figure 28 illustrates the

routes of two AMRs and one picker, where AMRs start the pick process from the depot and the picker is initially located at pick location e_4. In this scenario, AMR 1 handles order $o_1 = \{e_1, e_2\}$, arriving at time $t_1 = 0$, while AMR 2 manages order $o_2 = \{e_3\}$, arriving at time $t_2 = 5$. Assume equal speeds for both AMRs and pickers, with one time unit required to traverse to adjacent pick locations and cross aisles (travel time between two adjacent nodes in Figure 28a), and 12 time units for item retrieval from shelves. Set $\mathcal{T}$ shows the travel time between the locations in this example. $\mathcal{T} = \{\text{Depot} - e_1 = 5, \text{Depot} - e_2 = 11, \text{Depot} - e_3 = 5, e_1 - e_2 = 12, e_1 - e_3 = 0, e_2 - e_3 = 12, e_4 - e_1 = 2\}$



(a) Warehouse layout (b) Swarm policy (c) System-directed policy

Figure 28: Warehouse layout and comparison of picker routes and waiting times under swarm, system-directed, and manual picking

Under the swarm policy, AMR 1 arrives at location e_1 at time 5 (Figure 28b). After picking the item, the picker moves to location e_3, reaching there by time 17 to serve AMR 2. Once tasks at location e_3 are completed, the picker proceeds to location e_2, arriving at time 41. After picking up all the items, AMR 1 arrives at the depot at time 64. In the system-directed policy, the AMRs follow a similar efficient pattern: first AMR 1 reaches location e_1 at time 5, then transitions to location e_2 by time 29, and then to the depot. The picker, after picking item from location e_2, travels to location e_3 to serve AMR 2 and arrives there by the time 53. Finally, at time 70 AMR 2 reaches the depot (Figure 28c). For the manual picking policy, the picker arrives at location e_1 at time 5, moves onward to location e_2 by time 29, makes a stop at the depot by time 52, continues to location e_3 by time 57, and eventually returns to the depot by time 74. This example demonstrates that although AMRs follow the same route in both policies, the

picker travels 12 distance units less in the swarm policy compared to the system-directed policy by switching between AMRs while picking orders (14 units of travel distance in swarm policy versus 26 units in system-directed). Additionally, the total waiting time for the picker at pick locations is 12 units for AMR 1 in the swarm policy and 0 units in the system-directed policy. Conversely, AMR 2 has total waiting times of 12 units in the swarm policy and 48 units in the system-directed policy. Finally, the makespan in the swarm policy is 6 time units shorter than in the system-directed policy.

□

S2.2 Additional figures showing validation steps

Here, the additional figures related to the validation steps are presented. Figure 29 illustrates the conceptual model of the detailed discrete-event simulation. In this model, after every event, the simulation time undergoes an update. Once the simulation time reaches the designated termination time, the simulation run concludes.

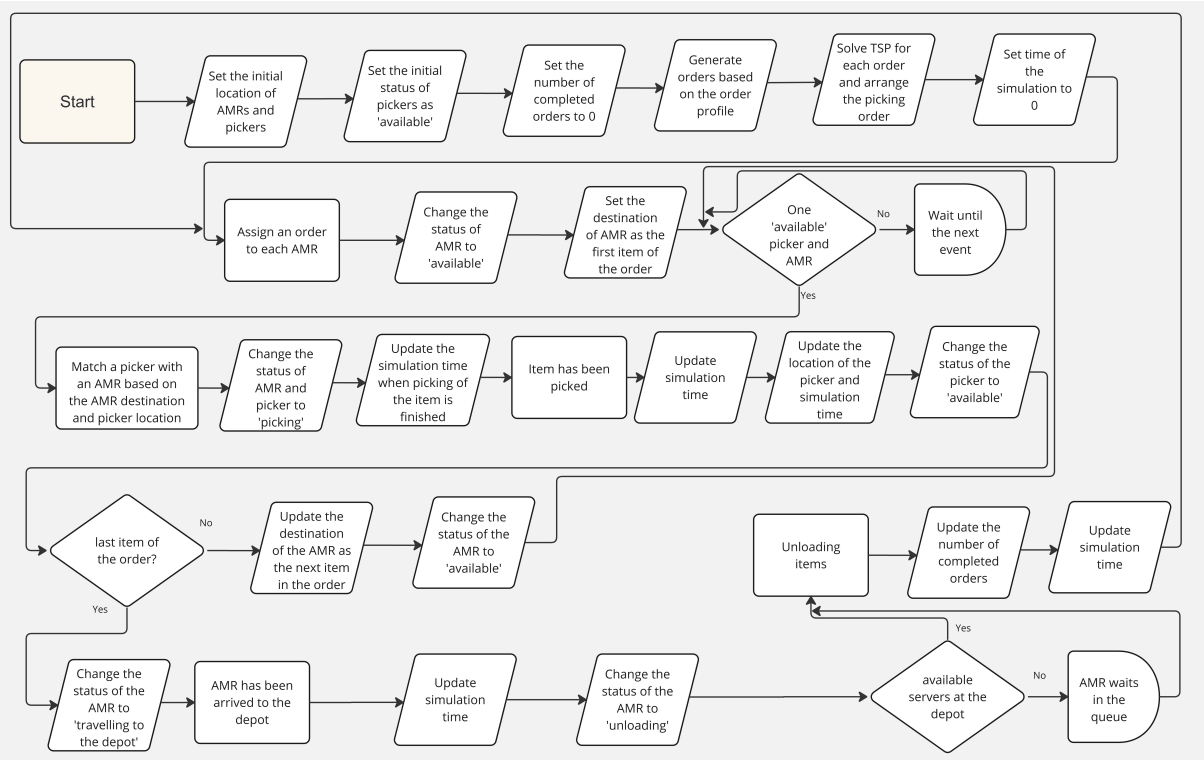


Figure 29: Conceptual model of the discrete-event simulation used for validation

Figure 30 provides a histogram of the order throughput errors in percentages.

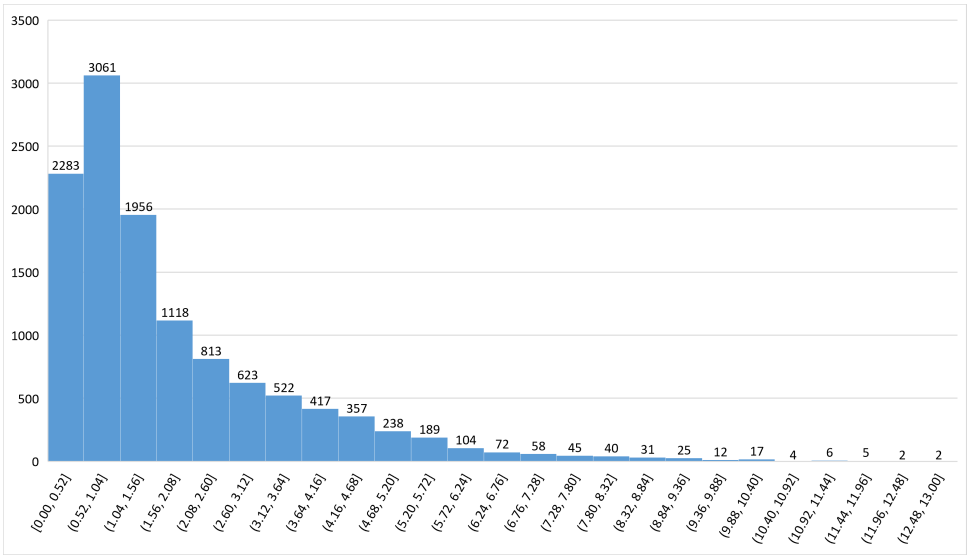


Figure 30: Histogram of throughput errors between analytical and simulation models

S2.3 Implementation of closest matching

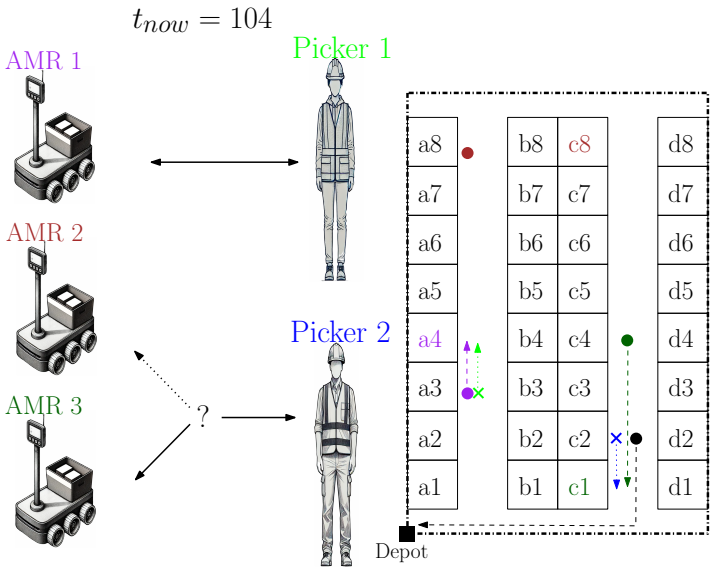
To implement the closest matching in simulation or reality, the state of each picker and AMR must be tracked, as well as the pick positions. A system state at a given point in time is represented by eight tuples, consisting of four tuples for AMRs and four tuples for pickers. These tuples are: s^r, the origin location of each AMR (e.g., the i-th element in s^r represents the origin of AMR i); e^r, the destination of each AMR; t^r, the time at which each AMR is expected to reach its destination; o^r the order assigned to each AMR; s^p, the; t^p, the time at which each picker is expected to arrive at their destination; r^p, the AMR assigned to each picker. We illustrate the closest matching with the following example.

A closest matching example: Consider a time stamp, say time $t = 104$, during the simulation, which involves four AMRs and two pickers as shown in Figure 31. Assume AMR 1 is matched with Picker 1, and they are en route from their origins at location $a2$ to pick an item at location $a4$. Meanwhile, Picker 2 has just completed picking an item for AMR 4 and now this AMR is heading to the depot. With Picker 2 free and available, it must be matched with either AMR 2 or AMR 3 which are waiting for a picker to be matched with.

The destinations of AMRs 2 and 3 are $c8$ and $c1$, respectively. Since $c1$ is closer to Picker 2's current location, Picker 2 is matched with AMR 3. As a result, AMR 3 will arrive at $c1$ by time 110, and Picker 2 will reach $c1$ by time 105. Therefore, for a matching, just the current location of available pickers, and the destination of AMRs are needed. These two tuples are highlighted in red in Figure 31. By maintaining and updating these tuples along with the current simulation

time, the system can consistently identify the timing and nature of the next event. Additionally, based on the orders assigned to AMRs, their next destinations can be determined. The pre-solved TSP routes dictate the sequence and locations for AMR item pickups.

Figure 31 illustrates this example, showing the state of the tuples before and after the matching process.



Before matching	After matching
$s^r = (a2, a8, c4, c2)$	$s^r = (a2, a8, c4, c2)$
$e^r = (a4, c8, c1, depot)$	$e^r = (a4, c8, c1, depot)$
$t^r = (105, -, -, 109)$	$t^r = (105, -, 110, 109)$
$o^r = (o2, o1, o3, o6)$	$o^r = (o2, o1, o3, o6)$
$s^p = (a2, c2)$	$s^p = (a8, c2)$
$e^p = (a4, -)$	$e^p = (a4, c1)$
$t^p = (105, -)$	$t^p = (105, 105)$
$r^p = (r1, -)$	$r^p = (r1, r3)$

Figure 31: Example of closest matching implementation in simulation

S3 Regression Models for the Service Rates of the Swarm Queuing Models, α_x, q_x, β_x, and μ_t^{-1}

This appendix provides the regression models, with α_x, q_x, β_x, and μ_t^{-1} as dependent variables, and speed ratios (S), item allocation method (nominal), number of aisles (A), number of blocks (B), order size (O), and the difference between the number of pickers and AMRs in the aggregated node x as independent variables. Table 11 shows the different considered levels of independent variables. The data are obtained from Monte Carlo simulation. Item allocation I is

coded with values 0 (random), 1 (ABC), and 2 (CBABC storage). Linear regression models are used for α_x, β_x, and μ_t, while a Logit regression model is used for q_x. Table 12 shows the parameters of these regression models along with the R^2 values and standard errors.

Table 11: Independent variables considered in regression models for service rates

Parameter	# of settings	Details
#Aisles	5	2, 4, 6, 8 and 10 aisles
#Blocks	3	1, 2, 3 blocks
Item assignment	3	Random, ABC and CBABC storage
x	25	{-8, -7, -6, ..., 0, 1, ..., 16}
Order size	5	2, 4, 6, 8, 10
AMR-to-picker speed ratio	2	Speed ratios of 1 ($v_r = 1$, $v_p = 1$) and 2 ($v_r = 1.33$, $v_p = 0.67$)

Table 12: Regression results for service rates and waiting probabilities of the swarm model (significance levels: *p<0.05, **p<0.01,***p<0.001), standard errors stated in parentheses. N = 11250 for α_x, β_x, and q_x. N = 450 for μ_t.

	Linear regression			Logit regression
	Dependent variable			Dependent variable
	α_x (in seconds)	β_x (in seconds)	μ_t (in seconds)	q_x
AMR-to-picker speed ratio	6.4584***(0.0997)	-6.3182***(0.0555)	-7.2426***(0.4714)	-0.7590***(0.0131)
Item allocation	-0.3725***(0.0611)	-0.2806***(0.0340)	1.0834***(0.2887)	-0.0147***(0.0080)
#Aisles	1.7032***(0.0176)	0.4443***(0.0098)	1.3956***(0.0833)	-0.0684***(0.0023)
#Blocks	4.7622***(0.0611)	4.3034***(0.0340)	7.4203***(0.2887)	-0.0444***(0.0080)
Order size (in lines)	-0.6610***(0.0176)	-1.3819***(0.0098)	-2.2098***(0.0833)	-0.1446***(0.0023)
x	0.3570***(0.0069)	-0.1166***(0.0038)	-	-0.0523***(0.0009)
Intercept	-7.1396***(0.2582)	23.6338***(0.1436)	25.3927***(1.2133)	2.4203***(0.0348)
Multiple R^2	0.6763	0.8220	0.8101	-
Adjusted R^2	0.6761	0.8219	0.8080	-
Pseudo R^2	-	-	-	0.4993

S4 An Example of the Equilibrium Equations for the Markov Chains of Swarm Picking and Complete Networks

This appendix presents the equilibrium equations for the Markov Chain of Swarm picking and complete networks presented in Figure 4.

S4.1 Swarm-picking network equilibrium equations

We can define the states of the aggregated node network shown in Figure 4a as $\eta = (\rho_\eta, \sigma_\eta, \tau_\eta)$ in which ρ_η, σ_η, and τ_η are the number of a matched AMR-picker at IS_1, IS_2, and IS_3 in this state, respectively. Therefore, the network of Figure 4a has $\frac{(l'+1)(l'+2)}{2}$ feasible states, where $l' = \text{Min}(x', P)$, and we define the set of these feasible states as $\mathcal{H}$. Now assume an example in which, $P = 2$ and $x' = 3$. It is noteworthy that there is always one AMR (in this example)

waiting in the synchronization station to be synced with a picker since the synchronization is instantaneous. Hence, the state of the synchronization station can be dropped from the state vector. In addition, we show the steady-state probability of these states as Π_{η_i}. Therefore, we have the following six feasible states:
$\mathcal{H} = \{\eta_1 = (0,0,2),\ \eta_2 = (0,1,1),\ \eta_3 = (0,2,0),\ \eta_4 = (1,0,1),\ \eta_5 = (1,1,0),$ and $\eta_6 = (2,0,0)\ \}$.

The equilibrium equations are as follows:

$$\Pi_{\eta_1} \times 2 \times \beta_{x=-1} = \Pi_{\eta_2} \times q_{x=-1} \times \alpha_{x=-1}$$
$$\Pi_{\eta_2} \times (\beta_{-1} + \alpha_{-1}) = \Pi_{\eta_3} \times 2 \times q_{-1} \times \alpha_{-1} + \Pi_{\eta_4} \times \lambda$$
$$\Pi_{\eta_3} \times 2 \times \alpha_{-1} = \Pi_{\eta_5} \lambda$$
$$\Pi_{\eta_1} \times 2 \times \beta_{-1} + \Pi_{\eta_2} \times (1 - q_{-1}) \times \alpha_{-1} + \Pi_{\eta_5} \times q_{-1} \times \alpha_{-1} = \Pi_{\eta_4} \times (\lambda + \beta_{-1})$$
$$\Pi_{\eta_2} \times \beta_{-1} + \Pi_{\eta_3} \times 2 \times (1 - q_{-1}) \times \alpha_{-1} + \Pi_{\eta_6} \times 2 \times \lambda = \Pi_{\eta_5} \times (\lambda + \alpha_{-1})$$
$$\Pi_{\eta_4} \times \beta_{-1} + \Pi_{\eta_5} \times (1 - q_{-1}) \times \alpha_{-1} = \Pi_{\eta_6} \times 2 \times \lambda$$
$$\sum_{i=1}^{6} \Pi_{\eta_i} = 1$$

S4.2 Swarm-complete network equilibrium equations

States of the aggregated network shown in Figure 4c can be defined as $\gamma = (x_\gamma, y_\gamma, z_\gamma)$ in which x_γ, y_γ, and z_γ are the number of AMRs at AGG_1 node, depot, and IS_4 in this state, respectively. With this definition, this network has $\frac{(R+1)(R+2)}{2}$ feasible states, and we define the set of these feasible states as Γ. In addition, we show the steady-state probability of these states as Π_γ. Assume the mentioned network with three AMRs and two servers at the depot ($R = 3$ and $m = 2$) Therefore, we have the following 10 feasible states:
$\Gamma = \{\gamma_1 = (0,0,3),\ \gamma_2 = (1,0,2),\ \gamma_3 = (2,0,1),\ \gamma_4 = (3,0,0),\ \gamma_5 = (0,1,2),\ \gamma_6 = (1,1,1),$
$\gamma_7 = (2,1,0),\ \gamma_8 = (0,2,1),\ \gamma_9 = (1,2,0),$ and $\gamma_{10} = (0,3,0)\}$.

The equilibrium probabilities for the states can be obtained by solving the balance equations for the continuous-time Markov chain. The equilibrium equations are as follows.

$$\Pi_{\gamma_1} \times 3 \times \mu_t = \Pi_{\gamma_2} \times u \times \delta_1$$

$$\Pi_{\gamma_2} \times (2 \times \mu_t + \times u \times \delta_1) = \Pi_{\gamma_3} \times u \times \delta_0 + \Pi_{\gamma_5} \times \mu_D$$

$$\Pi_{\gamma_3} \times (\mu_t + \times u \times \delta_0) = \Pi_{\gamma_4} \times u \times \delta_{-1} + \Pi_{\gamma_6} \times \mu_D$$

$$\Pi_{\gamma_4} \times u \times \delta_{-1} = \Pi_{\gamma_7} \times \mu_D$$

$$\Pi_{\gamma_1} \times 3 \times \mu_t + \Pi_{\gamma_6} \times u \times \delta_1 = \Pi_{\gamma_5} \times (2 \times \mu_t + \mu_D)$$

$$\Pi_{\gamma_2} \times 2 \times \mu_t + \Pi_{\gamma_7} \times u \times \delta_0 + \Pi_{\gamma_8} \times 2 \times \mu_D = \Pi_{\gamma_6} \times (\mu_t + \mu_D + u \times \delta_1)$$

$$\Pi_{\gamma_3} \times \mu_t + \Pi_{\gamma_9} \times 2 \times \mu_D = \Pi_{\gamma_7} \times (\mu_D + u \times \delta_0)$$

$$\Pi_{\gamma_5} \times 2 \times \mu_t + \Pi_{\gamma_9} \times u \times \delta_1 = \Pi_{\gamma_8} \times (\mu_t + 2 \times \mu_D)$$

$$\Pi_{\gamma_6} \times \mu_t + \Pi_{\gamma_{10}} \times 2 \times \mu_D = \Pi_{\gamma_9} \times (2 \times \mu_D + u \times \delta_1)$$

$$\Pi_{\gamma_8} \times \mu_t = \Pi_{\gamma_{10}} \times 2 \times \mu_D$$

$$\sum_{i=1}^{10} \Pi_{\gamma_i} = 1$$

S5 An Example of Solving the Equilibrium Equations for the Markov Chains of Swarm Picking and Complete Networks

In this appendix, equilibrium equations are first solved parametrically for a small example. Then, a slightly larger problem instance is presented, and the steady-state probabilities for the network states are found by numerically solving the equilibrium equations, while also employing Equations 2, 3, and 4 for comparison.

S5.1 An example of solving the equilibrium equations parametrically

In this subsection, the equilibrium equations of a small problem instance are solved parametrically. Assume the example provided in Online Supplementary S4 with three AMRs, two pickers, and two servers at the depot.

Solving equilibrium equations for the Markov chains of the Swarm-picking network: Assume the case in which all three AMRs are at the AGG node. Therefore, the equilibrium equations would be those presented in Online Supplementary S4.1. By solving the emerged equi-

librium equations parametrically we have:

$$\Pi_{\eta_1=(0,0,2)} = \frac{q_{-1}^2\alpha_{-1}^2\lambda^2}{(\alpha_{-1}\beta_{-1}+q_{-1}\alpha_{-1}\lambda_{-1}+\beta_{-1}\lambda)^2},$$
$$\Pi_{\eta_2=(0,1,1)} = \frac{2q_{-1}\alpha_{-1}\beta\lambda^2}{(\alpha\beta_{-1}+q_{-1}\alpha_{-1}\lambda+\beta_{-1}\lambda)^2},$$
$$\Pi_{\eta_3=(0,2,0)} = \frac{\beta_{-1}^2\lambda^2}{(\alpha_{-1}\beta_{-1}+q_{-1}\alpha_{-1}\lambda+\beta_{-1}\lambda)^2},$$
$$\Pi_{\eta_4=(1,0,1)} = \frac{2q_{-1}\alpha_{-1}^2\beta_{-1}\lambda}{(\alpha_{-1}\beta_{-1}+q\alpha_{-1}\lambda+\beta_{-1}\lambda)^2},$$
$$\Pi_{\eta_5=(1,1,0)} = \frac{2\alpha_{-1}\beta_{-1}^2\lambda}{(\alpha_{-1}\beta_{-1}+q_{-1}\alpha_{-1}\lambda+\beta_{-1}\lambda)^2},$$
$$\Pi_{\eta_6=(2,0,0)} = \frac{\alpha_{-1}^2\beta_{-1}^2}{(\alpha_{-1}\beta_{-1}+q\alpha_{-1}\lambda+\beta_{-1}\lambda)^2}.$$

Therefore:

$$\Pi_{(IS_1=0|3,2)} = \frac{(q_{-1}^2\alpha_{-1}^2+2q_{-1}\alpha_{-1}\beta_{-1}+\beta_{-1}^2)\lambda^2}{(\alpha_{-1}\beta_{-1}+q_{-1}\alpha_{-1}\lambda+\beta_{-1}\lambda)^2},$$
$$\Pi_{(IS_1=1|3,2)} = \frac{2\beta_{-1}(q_{-1}\alpha_{-1}^2+\alpha_{-1}\beta_{-1})\lambda}{(\alpha_{-1}\beta_{-1}+q_{-1}\alpha_{-1}\lambda+\beta_{-1}\lambda)^2},$$
$$\Pi_{(IS_1=2|3,2)} = \frac{\alpha_{-1}^2\beta_{-1}^2}{(\alpha_{-1}\beta_{-1}+q_{-1}\alpha_{-1}\lambda+\beta_{-1}\lambda)^2},$$
$$\Pi_{(IS_2=0|3,2)} = \frac{(\alpha_{-1}^2\beta_{-1}^2+2q_{-1}\alpha_{-1}^2\beta_{-1}\lambda+q_{-1}^2\alpha_{-1}^2\lambda^2)}{(\alpha_{-1}\beta_{-1}+q_{-1}\alpha_{-1}\lambda+\beta_{-1}\lambda)^2},$$
$$\Pi_{(IS_2=1|3,2)} = \frac{2\beta_{-1}\lambda(\alpha_{-1}\beta_{-1}+q_{-1}\alpha_{-1}\lambda)}{(\alpha_{-1}\beta_{-1}+q_{-1}\alpha_{-1}\lambda+\beta_{-1}\lambda)^2},$$
$$\Pi_{(IS_2=2|3,2)} = \frac{\beta_{-1}^2\lambda^2}{(\alpha_{-1}\beta_{-1}+q_{-1}\alpha_{-1}\lambda+\beta_{-1}\lambda)^2},$$
$$\Pi_{(IS_3=0|3,2)} = \frac{\beta_{-1}^2(\alpha_{-1}^2+2\alpha_{-1}\lambda+\lambda^2)}{(\alpha_{-1}\beta_{-1}+q_{-1}\alpha_{-1}\lambda+\beta_{-1}\lambda)^2},$$
$$\Pi_{(IS_3=1|3,2)} = \frac{2q_{-1}\beta_{-1}\lambda(\alpha_{-1}^2+\alpha_{-1}\lambda)}{(\alpha_{-1}\beta_{-1}+q_{-1}\alpha_{-1}\lambda+\beta_{-1}\lambda)^2},$$
$$\Pi_{(IS_3=2|3,2)} = \frac{q_{-1}^2\alpha_{-1}^2\lambda^2}{(\alpha_{-1}\beta_{-1}+q_{-1}\alpha_{-1}\lambda+\beta_{-1}\lambda)^2}.$$

Solving equilibrium equations for the Markov chains of the Swarm-complete network:

Based on the given example ($R = 3$ and $m = 2$) the equilibrium equations are equal to those presented in Online Supplementary S4.2. By solving the equilibrium equations parametrically

for this case, the steady-state probabilities corresponding to the network states can be found as follows:

$$\pi_{\gamma_1=(0,0,3)} = \frac{4u^3\delta_1\delta_0\delta_{-1}\mu_D^3}{\sum_{\gamma\in\Gamma} Z'_\gamma}, \quad \pi_{\gamma_2=(1,0,2)} = \frac{12u^2\delta_0\delta_{-1}\mu_D^3\mu_t}{\sum_{\gamma\in\Gamma} Z'_\gamma}, \quad \pi_{\gamma_3=(2,0,1)} = \frac{24u\delta_{-1}\mu_D^3\mu_t^2}{\sum_{\gamma\in\Gamma} Z'_\gamma},$$

$$\pi_{\gamma_4=(3,0,0)} = \frac{24\mu_D^3\mu_t^3}{\sum_{\gamma\in\Gamma} Z'_\gamma}, \quad \pi_{\gamma_5=(0,1,2)} = \frac{12u^3\delta_1\delta_0\delta_{-1}\mu_D^2\mu_t}{\sum_{\gamma\in\Gamma} Z'_\gamma}, \quad \pi_{\gamma_6=(1,1,1)} = \frac{24u^2\delta_0\delta_{-1}\mu_D^2\mu_t^2}{\sum_{\gamma\in\Gamma} Z'_\gamma},$$

$$\pi_{\gamma_7=(2,1,0)} = \frac{24u\delta_{-1}\mu_D^2\mu_t^3}{\sum_{\gamma\in\Gamma} Z'_\gamma}, \quad \pi_{\gamma_8=(0,2,1)} = \frac{12u^3\delta_1\delta_0\delta_{-1}\mu_D\mu_t^2}{\sum_{\gamma\in\Gamma} Z'_\gamma}, \quad \pi_{\gamma_9=(1,2,0)} = \frac{12u^2\delta_0\delta_{-1}\mu_D\mu_t^3}{\sum_{\gamma\in\Gamma} Z'_\gamma},$$

$$\pi_{\gamma_{10}=(0,3,0)} = \frac{6u^3\delta_1\delta_0\delta_{-1}\mu_t^3}{\sum_{\gamma\in\Gamma} Z'_\gamma}$$

Where:

$$\begin{aligned}\sum_{\gamma\in\Gamma} Z'_\gamma =& 4u^3\delta_1\delta_0\delta_{-1}\mu_D^3 + 12u^3\delta_1\delta_0\delta_{-1}\mu_D^2\mu_t + 12u^2\delta_0\delta_{-1}\mu_D^3\mu_t \\ &+ 12u^3\delta_1\delta_0\delta_{-1}\mu_D\mu_t^2 + 24u^2\delta_0\delta_{-1}\mu_D^2\mu_t^2 + 24u\delta_{-1}\mu_D^3\mu_t^2 \\ &+ 6u^3\delta_1\delta_0\delta_{-1}\mu_t^3 + 12u^2\delta_0\delta_{-1}\mu_D\mu_t^3 + 24u\delta_{-1}\mu_D^2\mu_t^3 + 24\mu_D^3\mu_t^3\end{aligned}$$

S5.2 An example of solving the equilibrium equations numerically

Now, we provide an example in which the steady-state probabilities corresponding to the network states are calculated both by solving forward equilibrium equations numerically and using Equations 2, 3, 4, and 8.

Assume a warehouse with the following characteristics and parameters: random item assignment, $A = 6$, $B = 2$, $S = 1$, $O = 6$, $P = 4$, and $R = 6$. In this case, the network input parameters are estimated as $\lambda^{-1} = 12$, $\mu_D^{-1} = 15$, $\mu_t^{-1} = 27.15055556$, and $u = 0.1667$. Table 13 shows the estimated values for α_x^{-1}, β_x^{-1}, and q_x

Equations 2, 3, and 4:

By defining the states of the queuing network shown in Figure 4a and solving the emerged equilibrium equations as explained in Online Supplementary S4.1, the steady-state probabilities for network states can be found. Table 14 presents the results for the aforementioned numerical example.

Table 13: Inputs of the queuing network for numerical equilibrium analysis

x	α_x^{-1}	q_x	β_x^{-1}
-2	18.82813	0.420466	17.55063
-1	21.17575	0.308976	16.62036
0	25.41057	0.078215	12.3967
1	22.09935	0.282654	13.91683
2	19.6195	0.414248	14.95668
3	17.73748	0.500504	15.88857

Table 14: Steady-state ((ρ, σ, τ)) probabilities of aggregated node states in Swarm-picking network

$x = 3 \rightarrow x' = 1,\ l' = 1$				
(0, 0, 1): 0.21099339	(0, 1, 0): 0.47061785	(1, 0, 0): 0.31838876		
$x = 2 \rightarrow x' = 2,\ l' = 2$				
(0, 0, 2): 0.02684459	(0, 1, 1): 0.17001179	(0, 2, 0): 0.26917911	(1, 0, 1): 0.10398539	(1, 1, 0): 0.32927947
(2, 0, 0): 0.10069965				
$x = 1 \rightarrow x' = 3,\ l' = 3$				
(0, 0, 3): 0.00110638	(0, 1, 2): 0.01864704	(0, 2, 1): 0.10475965	(0, 3, 0): 0.19618095	(1, 0, 2): 0.01012539
(1, 1, 1): 0.11376948	(1, 2, 0): 0.31958018	(2, 0, 1): 0.03088855	(2, 1, 0): 0.17353280	(3, 0, 0): 0.03140958
$x = 0 \rightarrow x' = 4,\ l' = 4$				
(0, 0, 4): 0.00000041	(0, 1, 3): 0.00004270	(0, 2, 2): 0.00167859	(0, 3, 1): 0.02932726	(0, 4, 0): 0.19214527
(1, 0, 3): 0.00002017	(1, 1, 2): 0.00158541	(1, 2, 1): 0.04154890	(1, 3, 0): 0.36295813	(2, 0, 2): 0.00037435
(2, 1, 1): 0.01962124	(2, 2, 0): 0.25710743	(3, 0, 1): 0.00308867	(3, 1, 0): 0.08094503	(4, 0, 0): 0.00955646
$x = -1 \rightarrow x' = 5,\ l' = 4$				
(0, 0, 4): 0.00032282	(0, 1, 3): 0.00532474	(0, 2, 2): 0.03293543	(0, 3, 1): 0.09054108	(0, 4, 0): 0.09333818
(1, 0, 3): 0.00301746	(1, 1, 2): 0.03732809	(1, 2, 1): 0.15392508	(1, 3, 0): 0.21157374	(2, 0, 2): 0.01057665
(2, 1, 1): 0.08722718	(2, 2, 0): 0.17984380	(3, 0, 1): 0.01647681	(3, 1, 0): 0.06794330	(4, 0, 0): 0.00962563
$x = -2 \rightarrow x' = 6,\ l' = 4$				
(0, 0, 4): 0.00139154	(0, 1, 3): 0.01420169	(0, 2, 2): 0.05435193	(0, 3, 1): 0.09245009	(0, 4, 0): 0.05896997
(1, 0, 3): 0.00905137	(1, 1, 2): 0.06928178	(1, 2, 1): 0.17676759	(1, 3, 0): 0.15033669	(2, 0, 2): 0.02207817
(2, 1, 1): 0.11266181	(2, 2, 0): 0.14372433	(3, 0, 1): 0.02393478	(3, 1, 0): 0.06106792	(4, 0, 0): 0.00973032

Then, we calculate $\pi_{(IS_j=r|x',P)}$ from the probabilities provided in Table 14, and compare the results with the results from Equations 2, 3, and 4. Table 15 shows the results of this comparison. As can be seen, the results of both approaches are identical.

The aggregated node throughput rates can be calculated using Equation 7 as:

$\delta(3) = 0.02653239$, $\delta(2) = 0.05288869$, $\delta(1) = 0.07887887$,

$\delta(0) = 0.10422044$, $\delta(-1) = 0.10440855$, $\delta(-2) = 0.10469127$.

Equation 8:

Now, we find the steady-state probabilities of the aggregated network states by using Equation 8 and by solving the equilibrium equations. The results are compared in Table 16. We use the values of $\delta(x)$ calculated in Online Supplementary S4.1 and other inputs provided in Online Supplementary S5.

Table 15: State probabilities of IS nodes in Swarm-picking network

		$\pi_{(IS_1=r\|x',4)}$		$\pi_{(IS_2=r\|x',4)}$		$\pi_{(IS_3=r\|x',4)}$	
x'	r	Equation 2	Equilibrium	Equation 3	Equilibrium	Equation 4	Equilibrium
1	0	0.68161129	0.68161124	0.52938213	0.52938215	0.78900658	0.78900661
1	1	0.31838871	0.31838876	0.47061787	0.47061785	0.21099342	0.21099339
2	0	0.46603537	0.46603548	0.23152963	0.23152963	0.69915838	0.69915823
2	1	0.43326492	0.43326487	0.49929126	0.49929126	0.27399707	0.27399718
2	2	0.10069970	0.10069965	0.26917912	0.26917911	0.02684456	0.02684459
3	0	0.32069402	0.32069402	0.07352988	0.07352990	0.72070360	0.72070350
3	1	0.44347505	0.44347505	0.30594929	0.30594933	0.24941761	0.24941768
3	2	0.20442135	0.20442135	0.42433984	0.42433983	0.02877241	0.02877243
3	3	0.03140958	0.03140958	0.19618099	0.19618095	0.00110638	0.00110638
4	0	0.22319431	0.22319422	0.01304006	0.01304006	0.90271197	0.90271232
4	1	0.40611263	0.40611260	0.10219440	0.10219438	0.09358638	0.09358606
4	2	0.27710295	0.27710301	0.30033494	0.30033492	0.00363837	0.00363835
4	3	0.08403366	0.08403371	0.39228537	0.39228538	0.00006287	0.00006287
4	4	0.00955645	0.00955646	0.19214524	0.19214527	0.00000041	0.00000041
5	0	0.22246223	0.22246226	0.04001934	0.04001937	0.56232489	0.56232466
5	1	0.40584436	0.40584437	0.19782325	0.19782331	0.34817003	0.34817015
5	2	0.27764766	0.27764763	0.36670429	0.36670431	0.08084008	0.08084017
5	3	0.08442012	0.08442011	0.30211489	0.30211483	0.00834218	0.00834220
5	4	0.00962563	0.00962563	0.09333823	0.09333818	0.00032282	0.00032282
6	0	0.22136529	0.22136523	0.06618628	0.06618619	0.42382878	0.42382924
6	1	0.40543746	0.40543743	0.25721337	0.25721320	0.40581440	0.40581427
6	2	0.27846427	0.27846431	0.37484385	0.37484386	0.14571214	0.14571189
6	3	0.08500267	0.08500270	0.24278661	0.24278678	0.02325313	0.02325306
6	4	0.00973032	0.00973032	0.05896989	0.05896997	0.00139155	0.00139154

Table 16: Steady-state probabilities of Swarm-complete network states

State	Equilibrium probability	Equation 8	State	Equilibrium probability	Equation 8
(0, 0, 6)	0.00000072	0.00000072	(1, 2, 3)	0.00011022	0.00011022
(1, 0, 5)	0.00003611	0.00003611	(2, 2, 2)	0.00138165	0.00138165
(2, 0, 4)	0.00075443	0.00075443	(3, 2, 1)	0.00774175	0.00774175
(3, 0, 3)	0.00845460	0.00845460	(4, 2, 0)	0.01641568	0.01641568
(4, 0, 2)	0.05378160	0.05378160	(0, 3, 3)	0.00000244	0.00000244
(5, 0, 1)	0.22766706	0.22766706	(1, 3, 2)	0.00006089	0.00006089
(6, 0, 0)	0.48057620	0.48057620	(2, 3, 1)	0.00050888	0.00050888
(0, 1, 5)	0.00000240	0.00000240	(3, 3, 0)	0.00142571	0.00142571
(1, 1, 4)	0.00009975	0.00009975	(0, 4, 2)	0.00000101	0.00000101
(2, 1, 3)	0.00166722	0.00166722	(1, 4, 1)	0.00001682	0.00001682
(3, 1, 2)	0.01401286	0.01401286	(2, 4, 0)	0.00007029	0.00007029
(4, 1, 1)	0.05942597	0.05942597	(0, 5, 1)	0.00000022	0.00000022
(5, 1, 0)	0.12578033	0.12578033	(1, 5, 0)	0.00000186	0.00000186
(0, 2, 4)	0.00000331	0.00000331	(0, 6, 0)	0.00000002	0.00000002

S6 Order Profiles

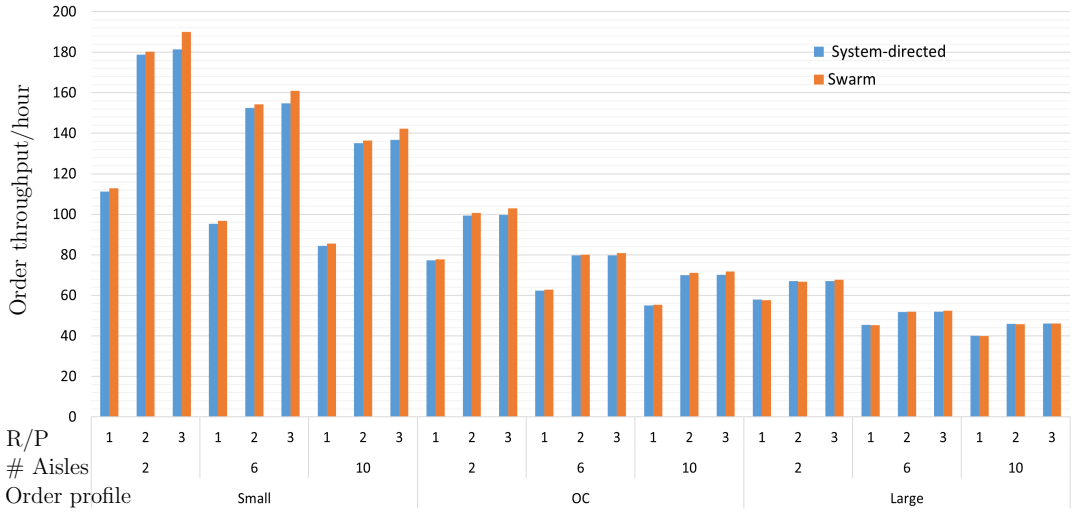
This section studies the effect of variable order sizes (i.e., the number of order lines). Three distinct order profiles are considered: small, large, and omnichannel, each reflecting a different order-size distribution. As shown in Table 17, each profile assigns a probability to the order size (from 1 to 12), allowing variability while preserving the dominant size characteristics of each profile. This modeling approach enables the analysis of systems under more representative demand patterns.

Table 17: Order-size profiles representing small, large, and omnichannel warehouse settings

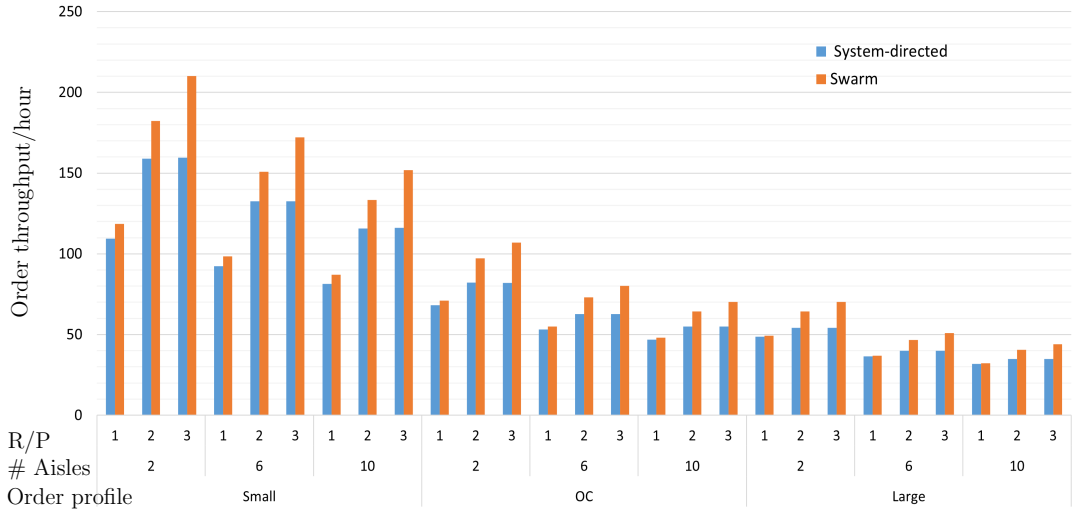
Order profiles	Small	Large	Omni channel
Order size		Probability	
1	0.60	0.01	0.35
2	0.20	0.01	0.10
3	0.12	0.03	0.08
4	0.03	0.05	0.03
5	0.02	0.08	0.03
6	0.02	0.10	0.05
7	0.01	0.10	0.06
8	0.00	0.12	0.05
9	0.00	0.20	0.06
10	0.00	0.15	0.10
11	0.00	0.08	0.04
12	0.00	0.07	0.05
Average size	1.77	8.00	4.76

The impact of order profiles on system performance is evaluated in terms of order throughput (per hour) under varying configurations, as shown in Figure 32. Results are presented for different speed ratios, and performance is compared between the swarm and system-directed policies. In most cases, higher throughput is observed under the swarm approach across all three order profiles. The largest relative gains are seen in the large-order profile with $S = 2$, where swarm achieves substantially greater throughput, indicating more efficient handling of high pick-density orders. However, when $S = 1$, the system-directed method occasionally yields higher throughput, particularly in the large-order profile with a small number of aisles, suggesting that under high congestion and limited flexibility, system-directed may lead to more effective coordination. Under the small-order and omnichannel profiles, swarm consistently outperforms or matches system-directed performance. These results suggest that while swarm is generally

more effective, especially when greater flexibility is available, it may be less advantageous in constrained environments with large, complex orders, and limited system resources.



(a) $S = 1$



(b) $S = 2$

Figure 32: Comparison of swarm and system-directed policies under heterogeneous order profiles with $P = 4, B = 2$, random item allocation, and TSP routing

S7 Dynamic matching

Both the system-directed and swarm policies can be further optimized using heuristic approaches. This section investigates the impact of Dynamic Matching (DM) on the throughput performance of the swarm policy through simulation. The DM strategy introduces two key differences compared to the previously discussed Closest Matching with pre-solved TSP (CM) routing strategy. First, instead of simply pairing each picker with the geographically closest available AMR—without regard for potential waiting times—DM aims to minimize coordination delays. Specifically, it

selects picker-AMR pairs that can arrive at the pick location sooner than other possible matchings, thereby allowing the picking process to begin as soon as possible.

Second, DM replaces the use of a pre-solved Traveling Salesman Problem (TSP) route for AMRs with dynamic routing. At each matching decision point, the AMR's next pick is chosen dynamically from the remaining items in the order, with the goal of minimizing coordination delays. It is outlined in Algorithm 3, while the baseline algorithm (CM) is outlined in Algorithm 2.

It is important to recognize that although this heuristic method enhance the system's performance, it does not guarantee a globally optimal solution. The current approach makes matching decisions based on local optimization at each step, without accounting for the broader, long-term impact on the system. In particular, the model lacks a mechanism for deferred matching—where agents (e.g., pickers or AMRs) might briefly wait instead of being immediately paired as soon as both resource types are available. Introducing a short waiting period could enable the system to better anticipate matching opportunities that arise shortly thereafter, potentially leading to more efficient and effective pairings. This kind of look-ahead strategy may help overcome the limitations of purely instantaneous matching, contributing to improved overall system performance.

Algorithm 2: Baseline matching (closest matching with pre-solved TSP)

Input: Set of available pickers $\mathcal{P}$, set of available AMRs $\mathcal{R}$,
current location of picker p $\mathcal{L}^p$, destination of AMR r $\mathcal{D}^r$

Output: Matched picker-AMR pair

matching distance $\leftarrow \infty$ `// Initialize matching distance as large value`

for *each picker* $p \in \mathcal{P}$ **do**
    for *each AMR* $r \in \mathcal{R}$ **do**
        if *distance*$(\mathcal{L}^p, \mathcal{D}^r) <$ *matching distance* **then**
            selected picker $\leftarrow p$;
            selected AMR $\leftarrow r$;
            matching distance $\leftarrow$ distance$(\mathcal{L}^p, \mathcal{D}^r)$;

Match selected picker with selected AMR;

Algorithm 3: Dynamic matching (DM; minimizing coordination delays with dynamic Routing)

Input: Set of available pickers $\mathcal{P}$, set of available AMRs $\mathcal{R}$, current location of picker p $\mathcal{L}^p$, current location of AMR r $\mathcal{L}^r$, set of remaining items for AMR r $\mathcal{O}^r$, current simulation time T

Output: Matched picker-AMR pair and updated destination

matching time $\leftarrow \infty$ `// Initialize matching time as large value`

for *each picker* $p \in \mathcal{P}$ **do**

  for *each AMR* $r \in \mathcal{R}$ **do**

    for *each item* $i \in \mathcal{O}^r$ **do**

      picker arrival $\leftarrow T + \frac{\text{distance}(\mathcal{L}^p, \mathcal{D}_i^r)}{\text{picker speed}}$;

      AMR arrival $\leftarrow T + \frac{\text{distance}(\mathcal{L}^r, \mathcal{D}_i^r)}{\text{AMR speed}}$;

      if max(*picker arrival, AMR arrival*) $<$ *matching time* **then**

        selected picker $\leftarrow p$;

        selected AMR $\leftarrow r$;

        selected item $\leftarrow i$;

        matching time $\leftarrow$ max(picker arrival, AMR arrival);

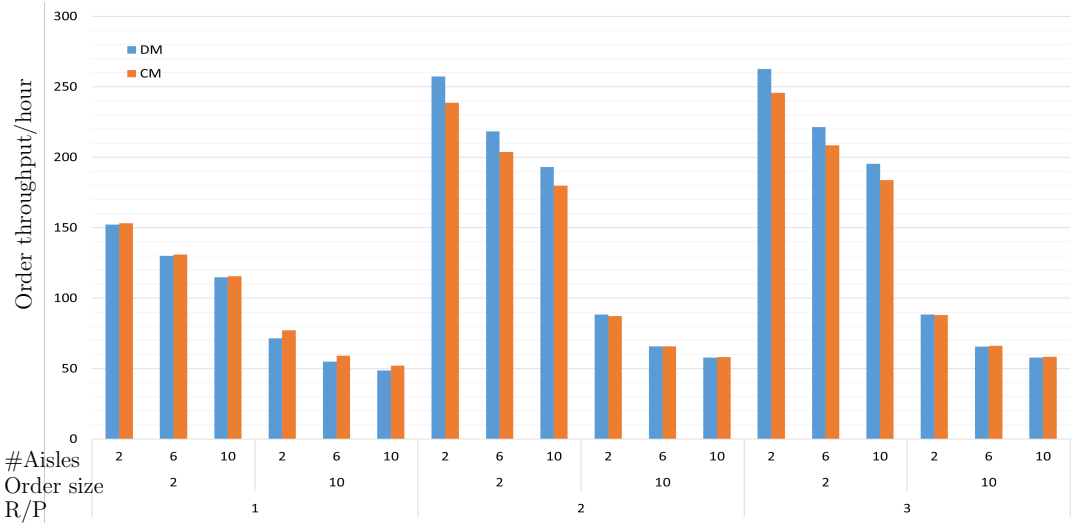
Match selected picker with selected AMR;

Set destination of AMR and picker to selected item;

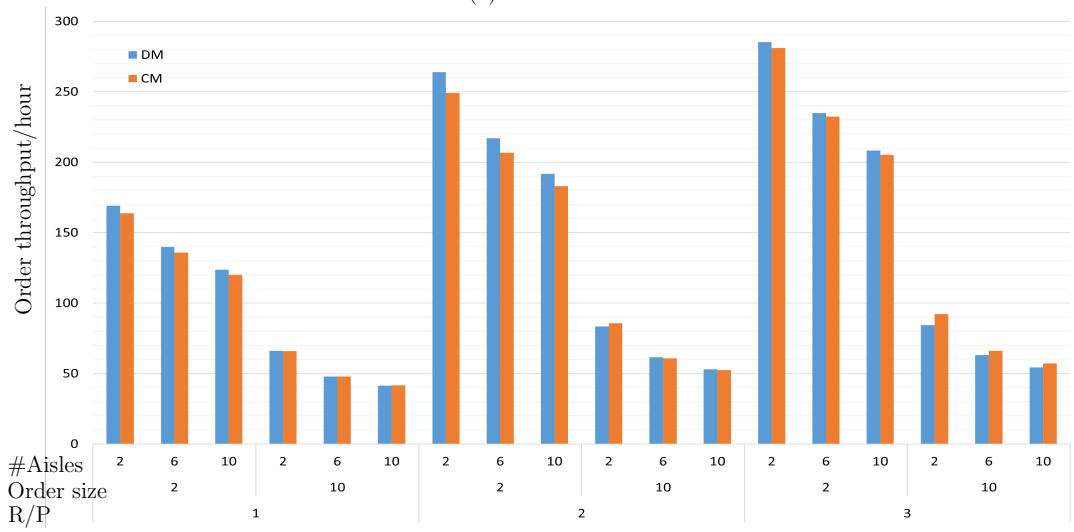
Figure 33 compares the order throughput of the two swarm policy variants: closest matching with pre-solved TSP (CM-TSP) and DM, in different R/P and speed ratios ($S = 1$ and $S = 2$).

When the R/P ratio is low, indicating AMRs are the bottleneck, CM-TSP generally outperforms DM. This phenomenon is observed because minimizing AMR travel distance becomes more critical under resource constraints, and CM-TSP is specifically optimized for this setting. However, as the R/P ratio increases, DM yields better performance. This outcome is observed because it prioritizes minimizing coordination delays, and therefore makes better use of limited picker capacity.

Overall, the results show that DM provides more adaptive and robust performance in picker-constrained environments, while CM-TSP remains preferable when AMR availability is the limiting factor.



(a) $S = 1$



(b) $S = 2$

Figure 33: Throughput comparison of closest matching with TSP routing versus dynamic matching where $P = 6, B = 2$

S8 Proof of Order Throughput Upper-bound for the Swarm Policy

This appendix provides the proof of Corollary 5. The network, as illustrated in Figure 3, can be simplified under the assumption of a sufficiently large number of AMRs within the network, guaranteeing a 100% utilization rate for all pickers. The synchronization station can be substituted with a G/M/P node given that the buffer of pickers remains consistently empty, where P represents the number of pickers, as indicated in Figure 34. This G/M/P node represents the average waiting time for an AMR to be matched with a picker. The depot, on the other hand,

operates as an G/M/m server. Consequently, the throughput of the network is ultimately governed by the rate of the slowest node.

Proof of Corollary 5: For a large number of AMRs and the picker travel times can be neglected. The throughput rate of the network depicted in Figure 34 establishes an upper bound for the throughput rate of the swarm policy. The throughput of this network is given by:

$$\mathrm{TH}_{SN} = \min(\lambda \times P \times u, \mu_D \times m)$$

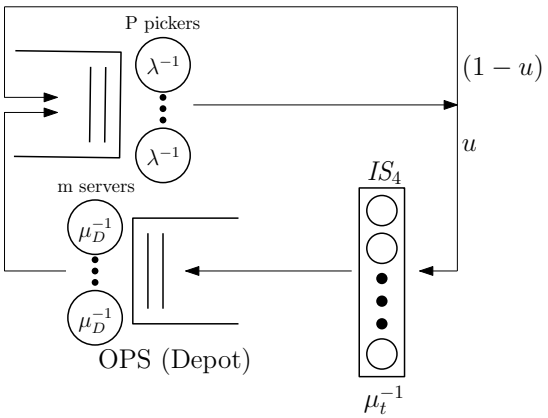


Figure 34: Simplified network representation showing throughput upper bound for swarm policy

□

S9 Cost Analysis Model

This appendix presents the model used in Section 6.3 for cost analysis.

Assume $TP^r(\xi, S, \upsilon, A, B, O, P, R)$ is the hourly order throughput of an AMR-assisted system with policy ξ, speed ratio S, item-to-storage allocation υ (either random, ABC, or CBABC), A aisles, B blocks, order size O, P pickers and R AMRs. In addition, assume $TP^m(\mathcal{S}, \upsilon, A, B, O, P)$ is the order throughput (in hours) of a manual picking system in which ψ is the speed of pickers and υ is the item allocation strategy. We show the total number of pick locations in a warehouse with Q, the number of working shifts with W, the hourly wage of the picker with c_p, and the daily lease cost of each AMR as c_r. Let us define the operational costs of an AMR-assisted system as $TC^r(\upsilon, A, B, P, R \mid Q, W, O, S, \xi)$ and a manual system as $TC^m(\upsilon, A, B, P \mid Q, W, O, \mathcal{S})$. Here, A represents the number of aisles, B the number of blocks, P the number of pickers, and R the

number of AMRs in AMR-assisted scenarios. The parameters include Q for the total number of required pick locations, W for working shifts, O for order size, S for the AMR-to-picker speed ratio (with $\mathcal{S}$ representing the speed of pickers in manual warehouses), and ξ for the operating policy in AMR-assisted scenarios. The following models find the minimum costs of a warehouse. Each of the solved problem instances is represented by a bar in Figures 22 and 23.

Model 1.

$$\min TC^r(\upsilon, A, B, P, R \mid Q, W, O, S, \xi) = \frac{(8c_P.P.W) + (c_r.R)}{TP^r(\xi, S, \upsilon, A, B, O, P, R)} \tag{14}$$

Such that:

$$60A.B = Q\ , \qquad \upsilon \in \{\text{random}, \text{ABC}, \text{CBABC}\}\ ,$$
$$A \in \{2, 4, 6, 8, 10\}\ , \qquad B \in \{1, 2, 3\}\ ,$$
$$P \in \{2, 4, 6, 8\}\ , \qquad R \in \{1P, 1.5P, 2P, 2.5P, 3P\}$$

Model 2.

$$\min TC^m(\upsilon, A, B, P \mid Q, W, O, \mathcal{S}) = \frac{(8c_P.P.W)}{TP^m(\mathcal{S}, \upsilon, A, B, O, P)} \tag{15}$$

Such that:

$$60A.B = Q\ , \qquad \upsilon \in \{\text{random}, \text{ABC}, \text{CBABC}\}\ ,$$
$$A \in \{2, 4, 6, 8, 10\}\ , \qquad B \in \{1, 2, 3\}\ ,$$
$$P \in \{2, 4, 6, 8\}$$