Simulation Modeling and Process Improvement Analysis for KitchenMaid Cabinet Manufacturing

Group 5: Jack Meghreblian, Palina Narkevich, Vernon Walker, Brody Kasprzak
Leeds School of Business, University of Colorado Boulder

Dr. Marco Better
April 22, 2026

Executive Summary

This project used discrete-event simulation in ExtendSim to analyze KitchenMaid's cabinet manufacturing process and evaluate ways to reduce total operating cost. KitchenMaid produces Royal, Elite, and Basic cabinets, each of which follows a different routing through a shared system with specialized labor, inspection, and possible rework. The goal was to minimize labor cost plus late-delivery penalties while keeping weekly labor cost at or below $20,000 and maintaining throughput of at least 80%.

The current-state analysis showed that the process is shaped by shared-resource dependencies, especially across labor pools that support multiple activities. The simulation modeled product-specific routing, stochastic arrivals, activity-time variability, inspection, and repair logic over a 90-workday horizon. Arrival data supported the use of an exponential interarrival-time distribution with a mean of 11.26 minutes, about 5.33 orders per hour. Optimization of the staffing configuration produced a recommended baseline of 3 Level 1 workers, 3 Level 2 workers, and 4 Level 3 workers, for a weekly labor cost of $17,550, which remained within the budget limit. Under this configuration, the system completed 5,009 of 5,016 created cabinets, yielding a throughput rate of 99.86%. Average cycle time was 79.02 minutes, about 10.02% of cabinets were late, and late-delivery penalties averaged about $15 per cabinet, contributing approximately $75,360 over the 90-day simulation period.

Several improvement scenarios were tested to determine whether the optimized current-state process could be improved further. The parallel-processing redesign performed substantially worse than the optimized baseline. Although it reduced staffing slightly, it sharply increased lateness, raised total penalties, and increased total operating cost. Removing rework by adding quality-control time earlier in the process also failed to improve performance, as it reduced completed output and slightly increased late fees and lateness rates. In contrast, the generalist labor model produced the lowest total operating cost, approximately $346,543, by replacing specialized labor pools with a single flexible worker type. While this scenario increased lateness exposure, the labor-cost savings were large enough to outweigh the additional penalties.

Overall, the analysis shows that simulation is an effective tool for evaluating KitchenMaid's production system and testing alternatives before implementation. The most practical recommendation is to use the optimized specialized staffing configuration, which satisfies the labor-budget and throughput requirements while keeping total cost relatively low. Although the generalist labor model achieved the lowest modeled cost, the optimized current-state staffing plan provides the best balance of realism, cost control, and operational performance.

Introduction

KitchenMaid is a national manufacturer of kitchen cabinets that produces three main standardized products: Royal, Elite, and Basic. Although these products are made in the same facility, each follows a different routing through the production process and places different demands on labor resources. The manufacturing system includes multiple processing steps, shared workers with different skill levels, and a quality inspection stage that may send some cabinets to rework before completion. Because of this complexity, system performance depends not only on product mix and arrival patterns, but also on staffing decisions, process flow, and the interaction between inspection, repair, and production capacity.

To better understand this system, KitchenMaid's process can be analyzed using discrete-event simulation. Simulation is appropriate because the process involves variability in production in order of arrivals, activity times, and repair needs, all of which affect cycle time, throughput, and cost. By modeling the current system in ExtendSim, it becomes possible to evaluate how the process performs under existing conditions and to test whether changes in staffing or process design could improve results.

Problem Statement

The purpose of this project is to analyze KitchenMaid's current manufacturing process and determine how to reduce total operating cost while maintaining acceptable system performance. Total operating cost includes both labor cost and late-delivery penalties. A cabinet is considered late if its cycle time exceeds 120 minutes, in which case the company incurs a penalty of \$200 for a Royal cabinet, \$120 for an Elite Cabinet, and \$50 for a Basic cabinet. In addition, management wants to keep the weekly labor budget at or below \$20,000 while maintaining an average throughput rate of at least 80%.

This project therefore focuses on two related questions. First, what staffing configuration minimizes average operating cost in the current process while satisfying the labor budget and throughput requirements? Second, can changes to the process design reduce cost even further beyond staffing adjustments alone? To answer these questions, the current system must be modeled and analyzed, and then alternative scenarios must be tested, including changes such as allowing Activity C to occur in parallel with Activities A or B and evaluating whether earlier quality control could reduce or eliminate the need for rework and final inspection.

Current State of Processes

This section maps the current process against the (part - a) flowchart, identifies structural bottlenecks, and establishes the baseline outputs from the first simulation interaction.

Process Flow Overview

The system follows a branch and conversion process. Royal cabinets enter at Activity A (L1+L2); Elite and Basic enter at Activity B (L2 only). All three types then converge at Activity C, Our guess at a primary bottleneck, before then diverging again by product type. Activity C is the only L1-exclusive activity that every order must pass through, concentrating all throughput pressure on the L1 pool at that single point.

Flow Path Analysis by Cabinet Type

Royal follows the longest production path and is the most operationally exposed; its dual-resource requirements at the start and end make throughput especially sensitive to staffing reductions. Given the $200 per late unit penalty, Royal also carries the highest financial risk. Elite has a shorter routing and avoids some processing steps, but it still shares key middle-stage activities with Royal. This means congestion from Royal can indirectly delay Elite as well. Basic bypasses several middle activities, but it converges with Royal at the final stage and competes for shared resources there. Because Basic represents 50% of all orders, even small delays can create meaningful total penalty costs despite its lower per-unit penalty.

Simulation Priorities, Resource Contention, Late Delivery Risk

Three areas present the greatest bottleneck risk. L1 is the most constrained resource tier because it supports multiple activities, including a required step for every cabinet, so any L1 reduction affects the full system. L2 has the broadest demand coverage, with the repair station posing the greatest risk because it permanently occupies one worker and can push defect-related delays into the wider process. L3 is required at multiple shared stages, so shortages can stall more than one step at once. Because work-in-process carries over daily, these conflicts can build into backlogs.

Late-delivery exposure increases the importance of these constraints. Cabinets exceeding the 120-minute cycle-time threshold incur penalties of $200 for Royal, $120 for Elite, and $50 for Basic. Royal is most vulnerable due to its long path, dual-resource requirements, and 7% failure rate. Elite has a shorter path but can still be delayed through shared stages with Royal. Basic has the lowest per-unit penalty, but its high volume makes total exposure meaningful; for example, a 10% late rate for Basic alone would generate about $2,500 in weekly penalties. With labor cost already exceeding the $20,000 budget by $7,250 (base), the system must balance both staffing cost and service performance.

These findings define three priorities for simulation: test L1 staffing at the shared required step, measure how the repair station reduces effective L2 capacity, and evaluate whether running the shared step in parallel with upstream work reduces the main bottleneck. The simulation will then move from baseline validation, to staffing optimization under the current design, and then to process redesign scenarios such as parallel processing and possible replacement of end-of-line inspection with in-process quality control.

Order Arrival Distribution Analysis (part-b)

To accurately model KitchenMaid's manufacturing system, the production-order arrival process was characterized using 300 observed interarrival times. The given sample had a mean of 11.26 minutes and a standard deviation of 10.94 minutes, producing a coefficient of variation of 0.97, which is close to the exponential distribution's theoretical value of 1.0. The data also showed positive skewness consistent with exponential behavior. A chi-square goodness-of-fit test using six bins produced a test statistic of 1.36, below the critical value of 11.07, so the null hypothesis of an exponential fit could not be rejected. Therefore, production-order arrivals were modeled in ExtendSim as an exponential distribution with a mean of 11.26 minutes, or approximately 5.33 orders per hour. This is important because an exponential distribution realistically reflects random order entry from a broad customer base and captures natural demand variability rather than assuming a fixed arrival schedule.

Baseline Simulation findings (part-c)

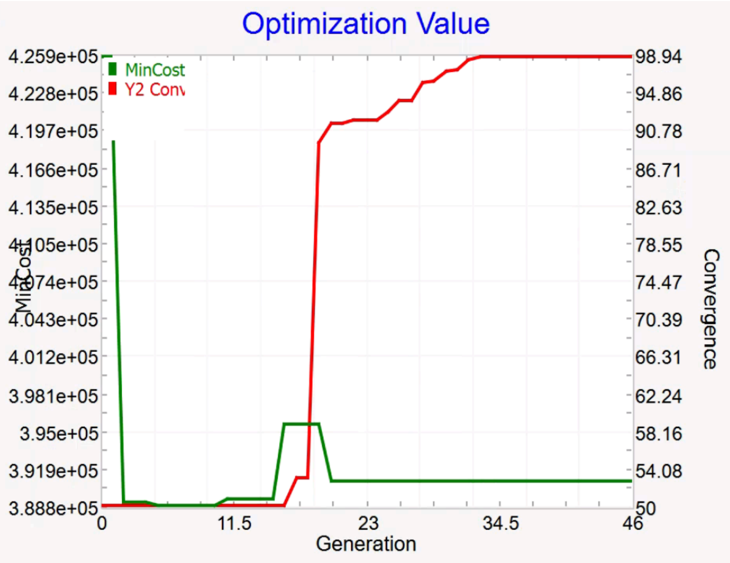
Using the process flow developed in Part A, the ExtendSim model was built to closely replicate the real-world production system. Each cabinet type (Royal, Elite, and Basic) was routed according to its specified activity sequence, with branching logic directing flow from Activities A and B into shared and product-specific paths. Resource pools were created for each labor tier (L1, L2, and L3), and all activities were configured to draw the appropriate resources based on required combinations. Failed units were routed to a repair station requiring a dedicated L2 resource before continuing through the remaining process steps without reinspection. This captures the interaction between inspection, rework, and production capacity. The arrival process was implemented using the exponential distribution identified in Part B, and product mix was assigned using probabilities of 20% Royal, 30% Elite, and 50% Basic. The system was modeled as a continuous process with work-in-process carrying over between days, and all simulations were run for 90 workdays to capture steady-state behavior.

Optimization Results Overview

To identify the optimal staffing configuration, an optimization routine was implemented to minimize total operating cost over 90 days with 5 runs per simulation round, defined as the sum of labor cost and late-delivery penalties, subject to throughput requirements. The initial optimization was run over 46 generations, as shown in *Figure 1* below. Total cost decreased from a little above $400,000 to a stable minimum of approximately $390,810. After roughly generation 20, improvements plateaued, indicating convergence to a near-optimal solution. The convergence metric approached 98.9%, confirming stability in the optimization results.

The optimized system processed 5,016 total arrivals, with 5,009 completed cabinets, resulting in a throughput rate of 99.86%, well above the required 80% threshold. The optimized weekly total cost was $21,711 ((135,000+94,500+86,400+75,360)/18 weeks) with a total operating cost of approximately $390,810 including late-delivery penalties over the 90 days. This number comes from $7,500 per week from L1 labor, $5,250 from L2 labor, and $4,800 from L3 labor. This puts total labor cost at $17,550 per week which is well below the $20,000 weekly budget. Of the total weekly cost $4,187 of it comes from weekly late fees. The final staffing configuration was: L1 = 3 workers and a utilization of 74%, L2 = 3 workers and a utilization of 50%, L3 = 4 workers with a utilization of 65%. What this means is that it is optimal for L2 to have 3 workers even though the utilization is low, it is better than not having enough in terms of cost.

Figure 1. *Minimum total Cost vs. Convergence*



Performance Outcomes

Under the optimized configuration, the system achieved an average cycle time of 79.02 minutes with a standard deviation of 32.5 minutes (Confidence interval - .90046, 90% of trials will contain this), remaining below the 120-minute late threshold on average. Late deliveries were relatively limited, with 10.02% cabinets observed as being late over the simulation. The average late cost between each of the three cabinet types was about $150 (502 cabinets) for each of the late cabinets, and out of all cabinets that made it through the average cost was $15. Which was a significant contributor to our total cost with a total contribution over the 90 days of $75,360. The tradeoff between increased labor or more late cabinets, and the associate fees, is something we paid close attention to in future variations of this simulation.

System Behavior Insights

The simulation results confirm key structural dynamics identified earlier. Activity C was not such a central constraint point, as we had first suspected. with an average queue length of 1 unit and average wait time of about 10 minutes, indicating moderate congestion at times but mostly clean flow through. The repair station processed 279 arrivals, all of which were completed, confirming that rework demand is non-negligible and continuously often draws on L2 capacity (pretty much always drawing on L2), this may be a spot for further exploration. This is one of the reasons why you need an extra L2 laborer even if utilization is lower in that labor type. We found that L1 is a labor bottleneck due to its importance and its high cost. The optimization model finds that it needs to make L1 as utilized as possible even if that means having an extra laborer somewhere else or slower processing in areas, as long as the latter two don't increase cost more than that.

Interpretation of Results

The optimized configuration demonstrates that reducing staffing levels can significantly lower labor costs while still maintaining high throughput performance. However, the results also show that system performance is sensitive to reductions in key resource pools, particularly L1, due to its role in shared and required activities. The obvious trade off is cost from being late for less workers, the optimization finds the best solution. As shown in the above findings the optimized simulation decreases cost while still meeting simulation requirements. However, we still see potential room for improvement. The model indicates that while staffing optimization provides meaningful cost reductions, further improvements will likely require process redesign. This motivates the analysis in Part D, where changes such as parallelizing Activity C or labor type redesign.

Model Improvements

These model improvements were tested because the original simulation showed that system cost was driven by a tradeoff between labor capacity, late-delivery penalties, and congestion at specific process points. In particular, the repair/rework loop, Activity C delays, and uneven labor utilization suggested that redesigning process flow or increasing labor flexibility could reduce bottlenecks and lower total operating cost. Furthermore, we decided to produce parallel processing, remove rework, and generalist labor models to address these pain points (please view *figures 2 & 3* below for visual comparison).

Parallel Processing Model

A third process redesign tested parallelization between Activities A–C and B–C. Compared with the optimized current-state model, this redesign performed worse overall. Total cost increased from $390,810 to $517,820, an increase of about $127,010 over the 90-day period. Although the optimized staffing configuration decreased from 10 total employees to 9 total employees, with staffing changing from L3 = 3 instead of 4, the lower labor expense was outweighed by a major increase in late-delivery penalties. The parallel processing model completed 4,926 cabinets, compared with 5,009 in the original optimized model. Late cabinets increased sharply from 502 to 1,921, raising the late percentage from 10.02% to 39%. As a result, total late cost increased from $75,360 to $213,360, meaning late fees made up about 41% of total system cost. Average cycle time also increased from 79.02 minutes to 122 minutes, moving above the 120-minute late threshold. This suggests that the added parallel-processing structure placed greater strain on labor resources, especially L1 (utilization 83%), and caused delays that outweighed any benefit from reducing one L3 worker.

Removing Rework

A process redesign was tested in which the rework loop was removed by adding quality control time to each activity before final inspection. This scenario was intended to reduce repair-related congestion, but the simulation results did not show improvement. Compared with the optimized current-state model, completed cabinets decreased from 5,009 to 4,921, a reduction of about 88 completed units over the 90-day period. Expected late fees also increased by approximately $4,000, and the proportion of late items rose from 10.02% to 11.1%.

From a statistical perspective, these differences were not statistically significant, the observed change is not strong enough to conclude that removing the rework loop reliably changes system performance. However, the direction of the results is unfavorable: lower throughput, higher late fees, and a higher late-item percentage. Overall, eliminating the rework loop does not appear to be an effective process improvement. The added quality-control time at earlier activities likely offsets any benefit from removing repair routing, suggesting that future improvements should focus instead on relieving small bottlenecks that occur.

Generalist Labor Model

A second process redesign tested a generalist labor model, where the system used one flexible employee type instead of separate L1, L2, and L3 labor pools. The optimization found 9 generalist employees should be used, when the cost per employee was set at $1775 per week . This scenario produced a total cost of $346,542.50, which is lower than the optimized original model of $390,810. This represents a cost reduction of about $44,286. The cost improvement comes mainly from reduced labor expenses. Although the model generated more late fees ($27,090 more late fees over 90 days), the savings from using fewer total employees outweighed the added penalty cost. In this scenario, 700 cabinets were late (14.08% of all finished cabinets), which included all cabinets that reached the end of the process. This model also produced a high expected labor utilization at 79%, suggesting that labor capacity was being used more efficiently. However, this also means the system had less slack, and there may have been periods where nearly all employees were occupied at once. Overall, the generalist model reduced total cost, but it did so by trading lower labor expense for higher lateness exposure. This iteration is an improvement in terms of total cost. Although this iteration is on the more impractical side, it does show that there can be value in training employees to do not just one task, but many.

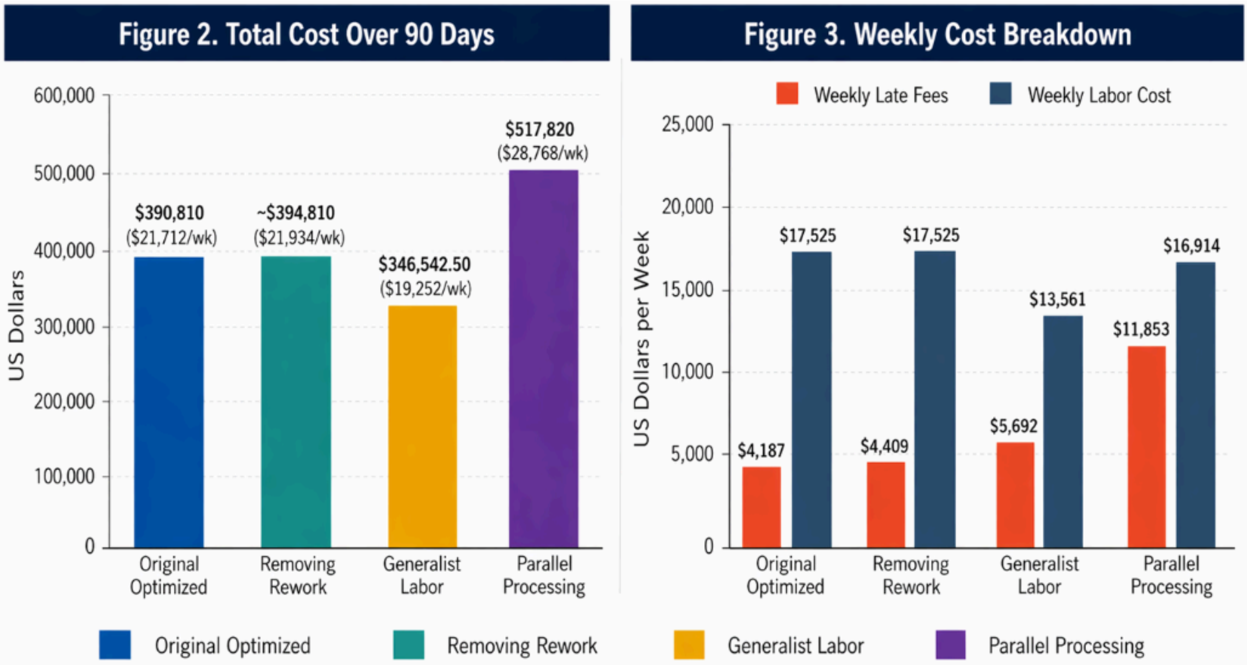


Figure 2. Total Cost Over 90 Days

Figure 3. Weekly Cost Breakdown

Conclusion

This analysis shows that discrete-event simulation can effectively evaluate KitchenMaid's production process. By modeling routing, shared labor, inspection, and rework, the simulation identified key cost and performance drivers. The optimization results show that staffing choices strongly affect total operating cost and can maintain high throughput when balanced carefully.

Additionally, the analysis highlights that identifying and addressing bottlenecks, particularly shared activities such as Activity C and resource-constrained labor pools is critical to improving system performance. While process modifications such as removing the rework loop were explored, results suggest that not all changes lead to improvements, reinforcing the importance of testing design decisions through simulation before implementation.

Several challenges were encountered during this analysis, especially the parallelization one where we struggled to find a good we to simulate parallel in extend sim, we created two random # generators and select the maximum of it to simulate what and when. Future work should focus on testing structural process improvements in more areas and further refinement. We would also try some batch size stuff, going into the bottle neck you want a low batch size but then things dependent on the bottleneck you want a bigger one to keep the model going. Overall, the findings confirm that optimization combined with targeted process adjustments is a valuable approach to improving operational efficiency.