

ACCA4019 Problem analysis and solution development

by Iwan Thomas

Submission date: 08-Jan-2026 12:07PM (UTC+0000)

Submission ID: 272523596

File name:

66124_Iwan_Thomas_ACCA4019_Problem_analysis_and_solution_development_692035_677824271.docx
(44.95K)

Word count: 1852

Character count: 10122

ACCA4019 Problem analysis and solution development

Planning

Must:

Reduce total travel distance - use routing methods + software

Allocate orders efficiently - use automated system + monitoring

No delivery is more than 30 mins late - avoid traffic + ensure vehicles are always available (don't overcap), find and deliver to earliest + most distant orders first, descend in order

Could:

Foolproof input system, ie no other countries, 0 orders, billion orders (minimal impact, dont spend time on this)

what about SHOULD and WOULD?

In order for an online grocer to maximise efficiency in its delivery routes, it first must be able to reduce total travel distance by finding the closest distribution hub, and only issuing orders to the recipients closest hub. This can be achieved by recording the recipients location in a delivery request process, then finding the distribution hub closest by utilising distance formula [x]

500 words on deconstruction

- Explain moscow method
- Present problems that need to be deconstructed
- Deconstruct problems as needed
- Describe how deconstructed problems are analysed

Planning on solution

Solution must:

Solve travelling salesman

Solve VRTTW

Allocate efficiently using waterfall

500 words on solution

- Explain flowchart solution
- Must be detailed
- Include decisions, assumptions and how the approach addresses each problem

Deconstruction

This written assessment will deconstruct the presented problems using the MoSCoW method primarily, amongst others. The MoSCoW method foremost involves the breaking down of features and solutions for a system by ranking them by desirability and impact, ranging from most impactful as a Must, and least impactful as a Wont have. From here, an analyst can assess the problems given. The solution the analyst must provide must be able to reduce total travel distance, as this would otherwise raise costs through fuel costs and maintenance, To be able to allocate orders efficiently, in order to maintain a system capable of smoothly operating without error, and finally no delivery being over 30 mins late, to avoid damaging relations between the distributor and the customer. All of these problems are requirements, and so must be filed as a must under MoSCoW. Some quality-of-life improvements that could be added but shouldn't be focused on would be ensuring the input system for deliveries is immune to logical errors, preventing users from inputting a nonsensical number of items, or a feature enabling the customer to give feedback to the distributor, to allow them to improve upon their system design.

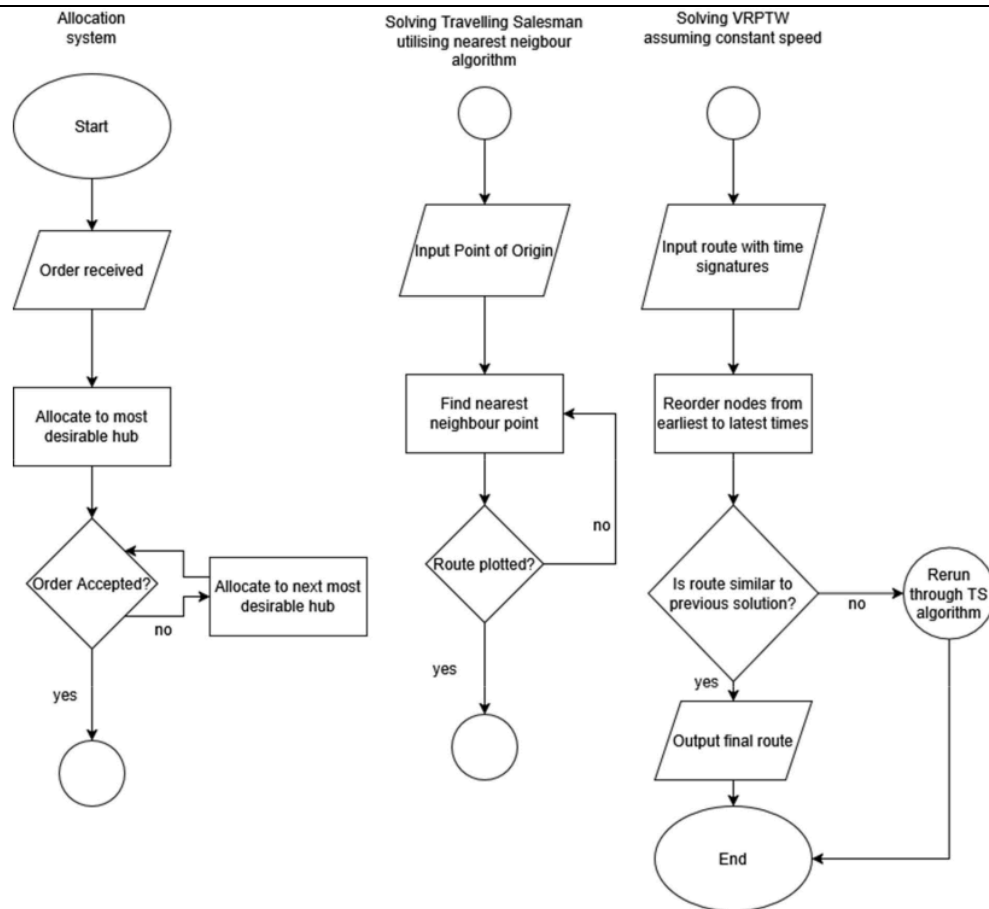
The first problem to be deconstructed, that the travel distance must be reduced, can be abstracted to finding the shortest route. The problem necessitates that the delivery agent must visit a set of stops in order, and then return to its origin point. This problem is often referred to as the 'Travelling Salesman Problem' [1]. Additionally it should be noted that this problem will exponentially increase in difficulty the more stops the set has.

The second problem to be constructed, that orders must be allocated efficiently, can be thought of as a chain of descending notes, with priority given to nodes found favourable by the allocator, being more reliable or in a more desirable location. Nodes given priority are moved towards the front of the chain. The allocator gives the first node the option of accepting or rejecting the order. If rejected, the request goes to the next node down the chain. This method is most successful when the highest priority nodes accept orders at the highest rate. This method is known as the 'waterfall method' [2]

The final problem to be deconstructed, that all orders are on time, and that no order is more than 30 mins late, can be thought of as the 'Vehicle Routing Problem with Time Windows' [3] In this problem, the analyst must find the most optimal route from a set of stops, where each of those stops has a set timeframe in which they must be delivered to, before the delivery agent returns to the point of origin. This can be assessed by ranking the times by the earliest windows, allowing the delivery agent to reach these stops in order of priority.

Solution

One possible solution for this problem is present within this diagram:



This diagram first attempts to solve the first part of the problem, that of efficiently allocated orders. Utilising the waterfall method [4], this solution attempts to allocate orders by ranking distribution hubs by preference. Preference can include factors such as, location, cost, reliability of hub and quality of service. The system then procedurally offers order requests until the order is accepted. As shown in the diagram, the system first starts when an order is received. The system then attempts to allocate the order to the most desired hub. This is then followed by a decision node, where the program will proceed to the next subsystem if the hub accepts. If the order is rejected however, the system will offer the order to the next hub in line. This process continues until a hub accepts the order, and the process moves on to the next stage. Here, the system must solve the Travelling Salesman Problem, in this case by using the nearest neighbour algorithm [5].

The system first asks the user for the point of origin, usually being the distribution hub that is handling the order. Utilising the algorithm, the system then finds the nearest point, and repeats this process until the entire route is plotted. Because the time to calculate Travelling Salesman using nearest neighbour exponentially increases the more nodes are added to the route, this solution assumes the distance being travelled is short, and the speed travelled at is constant. Otherwise the system is likely to take longer than desired in generating a route. Finally, the system must solve the Vehicle Routing Problem with Time Windows. Again, this subsystem assumes travel at a constant speed. This subsystem will first take the route generated before it, in addition to the time signatures that detail what times each node needs to be delivered to. The subsystem then reorders the nodes from the earliest to latest times, in order to ensure that deliveries are made on time. This outcome is then checked against the previous route, as if it has deviated significantly from the previous route order, it has

likely become inefficient in its routing, and so has to be run back through the Travelling Salesman algorithm before ending. Whether the route was reorganised or not, the system process then ends, and the final route is outputted to the delivery agent for them to complete the order.

Justification and Reflection

Among the solutions and problem solving techniques used in this assessment, The MoSCoW method is first used to establish what the solution requires, which then allows the problem to be subdivided. The problem was first subdivided into its constituent parts, that being the order must be allocated in a timely manner, the delivery route must be optimal and the deliveries must be on time. The problem was then abstracted into problems of a wider scope, which had been thoroughly covered by academic sources and research papers, that of the Travelling Salesman Problem, where an agent must path out the optimal route from a set of nodes, and then return to the point of origin, and the Vehicle routing Problem with Time Windows, where a route must be plotted that accounts for the time limits that individual nodes must be reached by. Additionally, the Waterfall method was also used as an abstraction of the problem requiring efficient allocation of orders. Waterfall method was found appropriate as it fit with the problems predefined scope and the method being simple to use for layman administrators. Both Travelling Salesman and Vehicle Routing Problem with Time Windows were deemed appropriate for the problem as they both matched the solution needed almost exactly.

On reflection of this assessment, there are several points where it could be improved. Firstly solving Travelling Salesman with nearest neighbour heuristics could cause problems, as nearest neighbour is a 'greedy' algorithm [7], meaning that the algorithm will solve routes by connecting the the immediate nearest node, rather than seeing the wider scope of the problem and finding the most efficient route that way. In some cases, the nearest neighbour algorithm can produce a route much longer than optimal [7]. Another issue with the proposed solution is the problems associated with using the waterfall method for order allocation. Because the waterfall method is an inherently rigid system, any later changes the designer wants to make to the system will lead to significant incurred costs both labour and financial [8] [4].

Additionally, the waterfall method assumes either that an agent will inevitably accept an order, or that there is an infinite or near infinite number as agents, as the method has no contingency if all agents reject the order other than repeating the request chain from the start, or cancelling the order. Finally, issues can occur when utilising Vehicle Routing Problem with Time Windows, for the assumption that travel speed is constant may not reflect as true when applied in the real world, possibly causing delivery delay due to variable speeds induced by outside factors such as road geography, traffic, and other unexpected incidents. Additionally, when this problem is solved using classic heuristics as in this assessment, rather than metaheuristics such as a genetic algorithm, the calculation and formula become increasingly complicated very quickly, meaning not only ever-increasing calculator times, but explanation of the solution to key layman organisers and administrator becomes near impossible. Using the Vehicle Routing Problem with Time Windows also doesn't account for other factors such as cargo capacity, fuel capacity, profit margin or delivery pickup, all of which are covered by other variants of a similar problem. These factors were not required in the project briefing, but are likely to occur in the real-world application.

References

[1]

T. Agarwal, "Vehicle Routing Problem (VRP): An Overview and Strategies," *Trackobit*, Dec. 12, 2023. <https://trackobit.com/blog/vehicle-routing-problem-decoded-what-why-and-how> (accessed Dec. 31, 2025).

[2]

F. Bolo, "Efficient inventory allocation: a key to retail success," *OneStock*, Jan. 29, 2024. <https://www.onestock-retail.com/en/blog/efficient-retail-inventory-allocation/> (accessed Dec. 31, 2025).

[3]

"Vehicle Routing Problem with Time Windows | OR-Tools," *Google Developers*. <https://developers.google.com/optimization/routing/vrptw> (accessed Jan. 4, 2026)

[4]

U. Senarath, "Waterfall Methodology, Prototyping and Agile Development," *Waterfall Methodology, Prototyping and Agile Development*, Jun. 2021, doi: <https://doi.org/10.13140/RG.2.2.17918.72001>. (accessed Jan. 4, 2026)

[5]

S. Goyal, "A Survey on Travelling Salesman Problem." Available: https://micsymposium.org/mics_2010_proceedings/mics2010_submission_51.pdf (accessed Jan. 4, 2026)

[6]

M. Andres Figliozzi, "The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 3, pp. 616–636, May 2012, doi: <https://doi.org/10.1016/j.tre.2011.11.006>. (accessed Jan. 4, 2026)

[7]

G. Gutin, A. Yeo, and A. Zverovich, "Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP," *Discrete Applied Mathematics*, vol. 117, no. 1–3, pp. 81–86, Mar. 2002, doi: [https://doi.org/10.1016/s0166-218x\(01\)00195-0](https://doi.org/10.1016/s0166-218x(01)00195-0). (accessed Jan. 5, 2026)

[8]

D. Parnas and P. Clements, "A RATIONAL DESIGN PROCESS: HOW AND WHY TO FAKE IT I. THE SEARCH FOR THE PHILOSOPHER'S STONE: WHY DO WE WANT A RATIONAL DESIGN PROCESS?" Available: <https://www.cs.tufts.edu/~nr/cs257/archive/david-parnas/fake-it.pdf> (accessed Jan. 5, 2026)



FINAL GRADE

81 / 100

GENERAL COMMENTS

Engagement with Literature 11/15

A clear and relevant reference list of 8 sources is included. Academic and professional sources are mixed appropriately. Concepts referenced in the body align with the cited literature but in-text citation use is inconsistent. Some references are used descriptively rather than critically engaged with.

Knowledge and Understanding 21/25

In general strong understanding and correct framing is demonstrated. Clear identification of core requirements as of assignment brief. Explicit use of MoSCoW with rationale. Problems are correctly mapped to recognised formulations: Travelling Salesman Problem and Vehicle Routing Problem with Time Windows. Understanding of why problems scale and become complex is demonstrated. However some sections are overly technical for Foundation level. Prior algorithmic knowledge is occasionally assumed. Variables are clearly implied and often explicit although there is no dedicated "Variables" subsection. Variables are linked directly to constraints and decisions. Awareness of computational complexity is shown. Some variables are embedded rather than clearly defined.

Cognitive and Intellectual Skills 16/20

Explicit reflection on nearest neighbour heuristic limitations, greedy algorithms, waterfall rigidity. It is acknowledged that chosen techniques may not be optimal. Evaluative thinking, not just description is demonstrated. Reflection is heavily technical; could link more explicitly to learning outcomes. Assumptions are clearly stated but some could be more clearly separated from limitations. Limitations are discussed realistically. Justification for method selection is explicit and reasoned.

Practical Application Skills 25/30

Multiple flowcharts included and they represent order allocation, TSP solving, VRPTW checking. Decision points, inputs, outputs are correctly used. Logical sequencing aligns with written explanation. Minor labelling clarity issues. Flowcharts are clearly explained in text and explanation links directly to decision points. It is clearly shown how each step addresses a specific problem constraint. Occasional verbosity.

Transferable Skills 8/10

Structured layout. Clear headings. Professional academic tone. Logical progression of ideas. But there is no separate title page and ToC. Minor grammatical and stylistic errors. Good professional standard

PAGE 1

Text Comment. what about SHOULD and WOULD?

Text Comment. ???

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6
