PATTERN - Pathfinding Algorithms to Tackle Effective Route Navigation

Tan Beng Yang, Goh Zi Yu, Ng Qi Heng and Chow Ban Hoe
River Valley High School
Singapore 649961
tan_beng_yang@students.edu.sg
goh_ziyu@students.edu.sg
ng_qi_heng_2@students.edu.sg
chow_ban_hoe@moe.edu.sg

Abstract

Food and parcel delivery routing is a major problem area due to its dynamism and complexity, exacerbated in recent years by growth in demand. The problem was modelled as a variation of the OP, a classical combinatorial optimisation problem, specifically the OTOP, involving a team of vehicles where each vehicle has a time budget to serve locations. A hybridisation of the metaheuristics Simulated Annealing and Iterated Local Search (SAILS) was constructed, able to escape local optima by sometimes accepting worse solutions. A website was also created to showcase the algorithm and computational tests were conducted on instances with varying parameters to simulate the large breadth of real-world scenarios. SAILS had higher profit than ILS and the initial solution and was able to acquire 58.8% of the profit increase over 20 minutes of runtime within 5 seconds, showing efficiency. However, the low improvement of SAILS over ILS shows SAILS has much room for improvement. Possible future work includes creating a more representative model with fewer assumptions and improving the algorithm through methods like reheating, genetic algorithms, and more precise parameter tuning. In short, a SAILS pathfinding algorithm was constructed to effectively and efficiently tackle delivery routing.

Keywords
Orienteering Problem, Team Orienteering Problem, Iterated Local Search and Simulated Annealing.

1. Introduction
The COVID-19 pandemic has accelerated the increase in demand for delivery routing, such as with 73% of Singaporeans spending more on food delivery services since circuit breaker started (Hirschmann 2020). The global market is expected to grow at a compound annual growth rate of 11.51 percent to reach 154.34 billion USD by 2023, from 107.44 billion USD in 2019 (Coppola 2022). The increased volume of orders has strained delivery riders, who have to make more deliveries in less time (Iskandar 2020) (Baertlein 2020). In addition, the number of delivery riders has increased. These two developments have greatly increased demand for delivery routing algorithms (McCourt n.d.), already difficult due to the dynamism and urgency of arriving orders (Van Lon et al. 2016). Such developments were already in effect before the pandemic due to convenience and workers' lack of time and only accelerated (Thompson 2019). Thus, such trends will continue post-pandemic (Gottfredson et al. 2021). Further applications have also arisen, with companies like Amazon exploring drone delivery. Thus, research in the area of delivery routing is likely to remain relevant for a long time.

Delivery service routing is considered a real application and extension of one of the classical combinatorial optimization problems, the Orienteering Problem (OP), a routing problem in which the goal is to determine a subset of locations to visit, and in which order, so that the total collected score is maximised and a given time budget is not exceeded (Gunawan et al. 2016). The unique case of delivery routing is considered a variation of the Open TOP (Open Team Orienteering Problem), an extension of the OP involving a team of vehicles where each vehicle has a time budget to serve locations (Gunawan et al. 2017) and is not required to return to its origin hub. To the extent of our knowledge, this study is the first that applies the OP and associated solutions specifically to the delivery routing scenario with its unique context. It is problem-specific.

1.1 Objectives

Apply the Open TOP to delivery routing, use Iterated Local Search and Simulated Annealing to create a solution, and model real-world conditions as accurately as possible.

2. Literature Review

Most food delivery companies currently employ a job-based system, where delivery riders are given a list of delivery requests from the surrounding area to choose from (Uber 2014). This results in inefficiency, as each driver only follows short-term interests and their own deliveries. As such, a delivery routing algorithm that determines paths the riders can greatly increase their efficiency by looking at orders on a macro level.

The OP models and its variants that are currently available are too general to be applied. For example, they do not account for variables like the presence of multiple hubs that delivery items originate from that must be visited before customers. Instead, most variants of the OP assume one start point which all vehicles must return to (Gunawan et al. 2016). This is not the case for food delivery riders who rarely return to their point of origin. Existing research also focuses on Operations Research, attempting to formulate a solution through mathematical modelling (Sun et al. 2020). However, such models are highly complex, and models that fully consider real-life conditions require too much computational power to be solved quickly (Rajgopal n.d.). In addition, some mathematical models are not prepared with actual data and thus have limited real-world coverage (Gera et al. 2018). Lastly, the OP and its variants has been proven to be NP (Non-Deterministic Polynomial-Time) -hard, meaning it is very time-consuming for the best solution to be found (Golden et al 1987). Rather, metaheuristics, less precise but able to return near-optimal solutions in a reasonable period of time are used.

Our proposed solution would use a hybridisation of Simulated Annealing (SA) and ILS (Iterated Local Search) to combine their strengths. ILS is simple and effective, but often gets trapped in local optima, where neighbours are not improving but the solution is not the best, as it usually only accepts better solutions (Gendreau et al. 1998, Marti et al 2018). Thus, SA, with the capability of accepting worse solutions, can help bridge this gap, while ILS provides a lower-level mechanism for SA to utilise. SAILS (Simulated Annealing and Iterated Local Search) has proven effective in other situations, with 50 new best-known solutions discovered on benchmark TOPTW (Team Orienteering Problem with Time Windows) instances and having efficient computational times (Gunawan et al. 2017). It was also applied with success to other problems such as the Inventory Routing Problem and Hub Location Problem, both demonstrating short computational times and discovering new optimal solutions. (Alvarez et al. 2018) (Davari et al. 2015).

Food delivery has been growing rapidly, creating strain for services and delivery riders alike. Yet, existing solutions remain unsuitable. Using SAILS for the routing of food delivery services, our research would allow delivery personnel to increase their efficiency and maximise profits, assisting the growth of the food delivery industry and enhancing the convenience and utility this industry provides to businesses and customers.

3. Methods

3.1 Data generation

Synthetic data is generated to provide a setting for the algorithms. A series of objects, namely nodes (delivery recipients), hubs (stores/restaurants) and vehicles (delivery riders) are randomly generated. Default variables are set to mimic real-world conditions as far as possible as shown in Table 1, with highlighted variables altered in different instances to simulate the wide variety of real-life situations (Quek 2020). We consider an ideal situation with flat terrain, straight pathways, and constant speed. In Figure 1 below, M refers to the set of all hubs while N refers to the set of all nodes. N^1 and N^* denote the sets of unscheduled and scheduled nodes respectively where ($N^1 \cup N^* = N$).

Table 1: Default Value of All Variables

Hub Number	Hub Waiting Time/min	Maximum distance of node from hub/km	Node Number	Node Waiting Time/min	Node Price Range/$	Vehicle Number	Vehicle Speed/ km h^{-1}	Vehicle Time Budget/h
20	1	5	200	1	2-20	20	10	5

3.2 Greedy construction heuristic

An initial simple pathway is constructed by representing each node and hub as a set of coordinates (x, y). Each vehicle is instructed to go to the nearest node ordering from a hub it has been to. Straight line distance between nodes is calculated to create a time matrix, as distance is a function of time and speed. If there are no nearby nodes, the vehicle will go to the nearest hub. The routing continues until the vehicle reaches its time budget. As a result, it can generate a good, but imperfect initial pathway which can be modified, reducing the computational time taken by later steps to optimise the solution.

In addition, the ideas behind this heuristic are similar to those employed by most food delivery companies: to allow delivery personnel to select from a list of available orders (Reyes et al. 2018), which would incentivise going to nearer nodes and induce them to wait near hubs when inactive. Hence, the greedy construction heuristic can provide a negative control to measure the results of SAILS, as well as an initial solution for SAILS to improve.

```
N* ←node 0
N¹ ← N\node 0
Initialise S0 ← N*
F ← UpdateF(N¹, M)
while F = ∅ do
        (n*, p*, m*) ← Select(F)
        S0 ← (n*, p*, m*)
        Update P(m)
        N¹ ← N¹ ₩ {n*}
        N* ← N* ∪ {n*}
        F ← UpdateF(N¹, M)
end while
return S0
```

```
S0 ← Greedy (N, M)
S0 ← LocalSearch(S0, N*, N¹, M)
S* ← S0
        S0 ← LocalSearch(S0, N*, N¹, M)
        if S0 better than S* then
                S* ← S0
        else
                S0 ← S*
        end if
end while
return S*
```

Figure 1: Pseudocode of Greedy Algorithm (Left) and ILS (Right)

3.3 ILS

The solution is then modified with ILS, a metaheuristic modification of traditional local search methods. A simple explanation is in Figure 2, with pseudocode in Figure 1. Unlike heuristics which drive towards local optima, metaheuristics aim to escape local optima to reach the global optimum. Let the greedy solution be S_0. It first uses local search on S_0 to find neighbouring solutions, moving towards better solutions until it reaches a local optimum. It then applies perturbation to escape local optima, and local search is carried out on the perturbation to produce a solution S'. Then, the acceptance criterion determines whether to accept S'. If accepted, S' replaces S_0 and the process of perturbation is repeated on S'. Otherwise, S' is discarded and the search returns to the previous solution. This iterative process reduces the likelihood of being trapped in local optima, offering a greater chance of reaching the global optimum (Glover et al. 2003). As one of the older and more general TOPTW solutions, it can also be hybridised with other more specific metaheuristics.

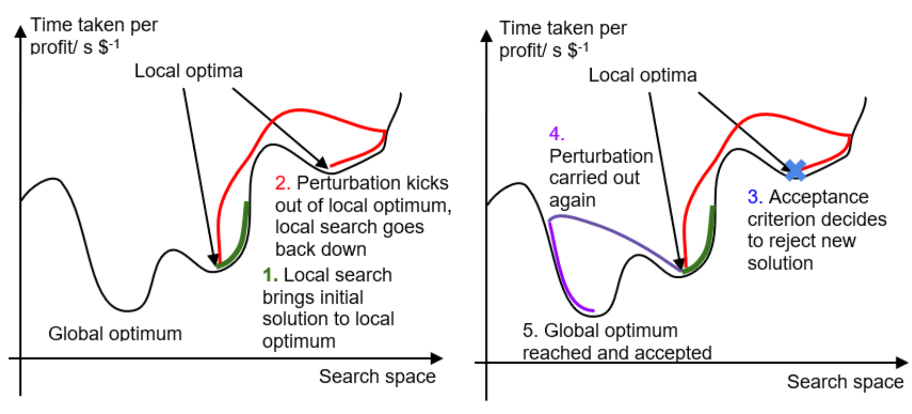

Figure 2: Illustration of ILS

Table 2: Local search operators

Operator	Description
Intra-vehicle Swap	Shuffles order of nodes and hubs
Inter-vehicle Swap	Swaps nodes between vehicles
Move	Moves a node from one vehicle to another
Insertion	Adds profitable but unserved nodes and hubs
Replacement	Replaces less profitable nodes with more profitable nodes

The above operators as shown in Table 2 are run consecutively, so that swap and move operators decrease travel time to allow insertion and replacement to add more nodes to increase total profit. Intra-vehicle swap removes all hubs and nodes in the vehicle's path except for the origin. Afterwards, hubs are added back in a random order, followed by nodes, ensuring nodes are visited after the hubs they order from. The new pathway is accepted only if time taken is reduced. This follows the logic of delivery routing where vehicles pick up objects first, distinct from other OP problems. The inter-vehicle swap extends this idea, exchanging nodes between vehicles when the nodes correspond to the same hub and if swapping saves time. The move operator iterates through every node of every vehicle, removing it and seeing if inserting into another path could reduce total time taken.

Insertion starts from the vehicle with the longest remaining time. Unserved nodes are sorted by profit, with more profitable nodes prioritised. If the node is not served by an existing hub, the hub will also be added in if possible. This serves to maximise profits while giving the widest range of options to select the best option. Replacement likewise starts from the vehicle with the most remaining time; this is to ensure the maximum number of replacements can be done as both operations increase the time taken. It then prioritises unserved nodes by profit and checks through the vehicles as to whether it can replace any currently served node. This process continues until no more replacements can be made. The set of local search operators is run for a fixed number of iterations.

Perturbation must not be too weak such that local search reverses it and the solution returns to the previous local minimum. Yet, it must also not be too strong until it behaves like an inefficient random restart (Glover et al. 2003). With this in mind, we introduce a new perturbation mechanism tuned to our setting, “remove chunks”, inspired by the Shake operator (Vansteenwegen et al. 2009). In this operator, hubs and their associated nodes in each route are classified into “chunks”. For each iteration of ILS, a certain number P of chunks is randomly removed from the entire solution, and if the solution S is not updated for T iterations, P is increased by 1. If S is updated, P is reset to its initial value. This makes sure hubs are served before their corresponding nodes. Additionally, starting with a weak perturbation and gradually increasing its strength ensures continual improvement without perturbation becoming too strong. An alternative method of removing chunks was proposed, where chunks are removed starting from the least profit per time to have a more targeted approach.

3.4 Simulated Annealing

The acceptance criterion determines the balance between intensification, which effectively exploits promising regions in the vicinity of good solutions, and diversification, which helps to escape from local optima (Soria-Alcaraz et al. 2019). Only accepting better solutions is efficient, but can get stuck in local optima when neither of the neighbouring local optima which are reached by perturbation is better. Thus, even with the increased search space of ILS, it may be insufficient to overcome local optima (Gendreau et al. 1998, Marti et al 2018). SA can be regarded as a modification of the acceptance criterion to accept worse solutions with a low probability, a form of diversification which makes it easier to reach global optima. Starting with a temperature T_0, as opposed to ILS which only accepts S' when better, in SA, if the new path is worse, it would have an $e^{(L1-L0)/T}$ chance of replacing the current path. After InnerLoop iterations, the temperature would decrease by a small amount known as ΔT, determined by a coefficient α. The value of InnerLoop determines how many times it is run before temperature decreases. After each decrease, the algorithm is run again for InnerLoop iterations, and as T decreases by ΔT, the probability of the final path changing decreases. The number of times this occurs is denoted as OuterLoop. Thus, SA can gradually converge on the global optimum (Aarts et al. n.d.). Pseudocode is shown in Figure 3 below.

```
S0 ← Greedy (Nodes, Hubs)
S* ← S0
S' ← S0
Temp ← T0
while TimeLimit has not been reached do
        InnerLoop = 0
        while InnerLoop < MaxInnerLoop do
                S0 ← Perturbation(S0, Unserved Nodes, Nodes, Hubs)
                S0 ← LocalSearch(S0, Unserved Nodes, Nodes, Hubs)
                δ ← obj value of S0 - obj value of S1
                if δ > 0 then
                        S' ← S0
                        if S0 is better than S* then
                                S* ← S0
                        end if
                else
                        r ← rand[0, 1]
                        if r < exp(δ/Temp) then
                                S' ← S0
                        else
                                S0 ← S'
                        end if
                        InnerLoop ← InnerLoop + 1
                end while
                Temp ← Temp × α
                S0 ← S*
                S' ← S0
end while
return S*
```

Figure 3: SA Pseudocode

3.5 Data representation

The data the algorithms produce is transformed using graphing software Plotly to represent hubs, nodes, and routes on a map. This makes it easier for delivery riders and those not specialised in programming to interpret. An interactive website at https://21rv02.pythonanywhere.com was created for users to create their own dataset, randomly generating hubs, nodes and vehicles based on their inputs. The user can then adjust input SAILS parameters to optimise algorithm performance to the dataset generated. It also contains a glossary of key terms and additional information on this team. A real-life instance was generated centred around our neighbourhood with hubs like the local shopping mall Jurong Point and nodes like housing block 354, accounting for road systems and traffic using a time matrix generated by the Bing Maps Application Programming Interface, using latitude and longitude coordinates. The instance was then uploaded onto the website presented using an embedded Google Maps, as shown in Figure 4, allowing the user to see specific routes and travel time estimates for each vehicle.

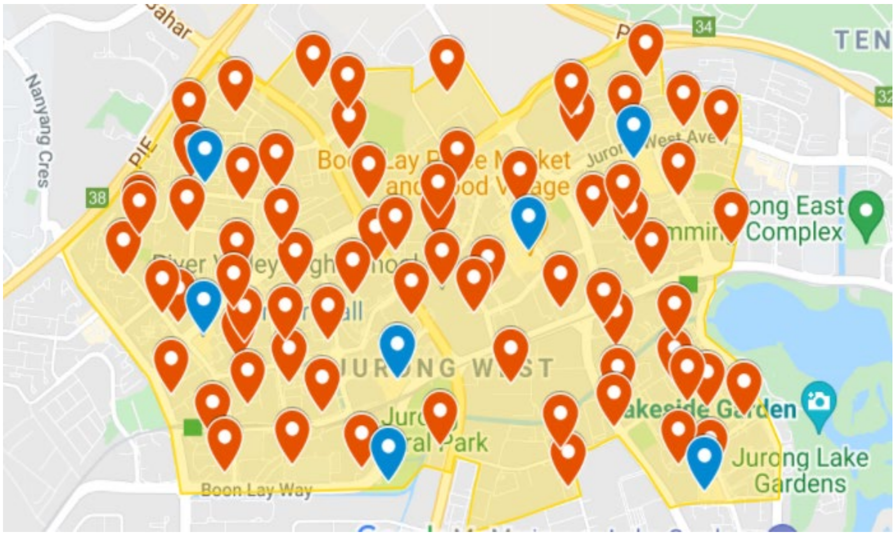

Figure 4: Representation of Hubs (Blue) and Nodes (Red) Selected in Real-Life Instance

4. Data Collection

A comparative study was then conducted between the two perturbation mechanisms. Different combinations of different values of T_0 (500, 750, 1000), α (0.6, 0.75, 0.9), P (2, 4, 6) and T (6, 8, 10) are used, so as to compare between the near-optimal results of the two perturbations. Each set of values is run on Big Data 10 to 15, the instances comparing hub to node ratio, three times each with 10 inner loops and 10 outer loops, as longer runtimes generally present diminishing returns and hub to node ratio contained greatest variation in profit increase in test runs, suggesting greatest diversity of instances. It was found that the random perturbation mechanism had superior performance.

Parameter tuning was then performed using the same set of parameters as above on every instance. Top five results from each instance were taken and compared, with the mode taken to generate a "best fit" parameter value able to accommodate as many instances as possible. The mode was used as the method of selection for the solution to fit the highest number of instances, given the high variation in outcomes and high possibility of outliers. We thus conclude the parameter values as: $T_0 = 750$, $\alpha = 0.9$, $P = 2$, $T = 8$.

Optimal parameter values were run five times on each instance, and the average taken. To test for theoretical potential, a long computational time limit of 20 minutes was given for each run to allow algorithms to try to reach their maximum potential. Data from this run was also organised according to instance variables to analyse their impact on results. Additionally, given the dynamic nature of food delivery routing which causes routes to change rapidly, it is important to run tests with short time limits to evaluate practical applicability of each algorithm. With 5-second intervals, time limits from 5 to 50 seconds were run with each algorithm on each instance 10 times, and the average taken.

5. Results and Discussion

5.1 Numerical Results

From Figure 5, it is concluded that random perturbation is slightly better than targeted perturbation, and it was thus used. The anomaly at 0.25 hub to node ratio is discounted as other instances had lower hub to node ratios, suggesting random perturbation would perform better for most datasets. The poor performance of targeted perturbation could be because having the least profit per chunk does not necessarily mean it should be removed; it is instead possible that these chunks were inserted by Local Search, and the repeated removal and insertion meant targeted perturbation was unable to improve further. In contrast, random perturbation would demonstrate greater ability to escape local optima.

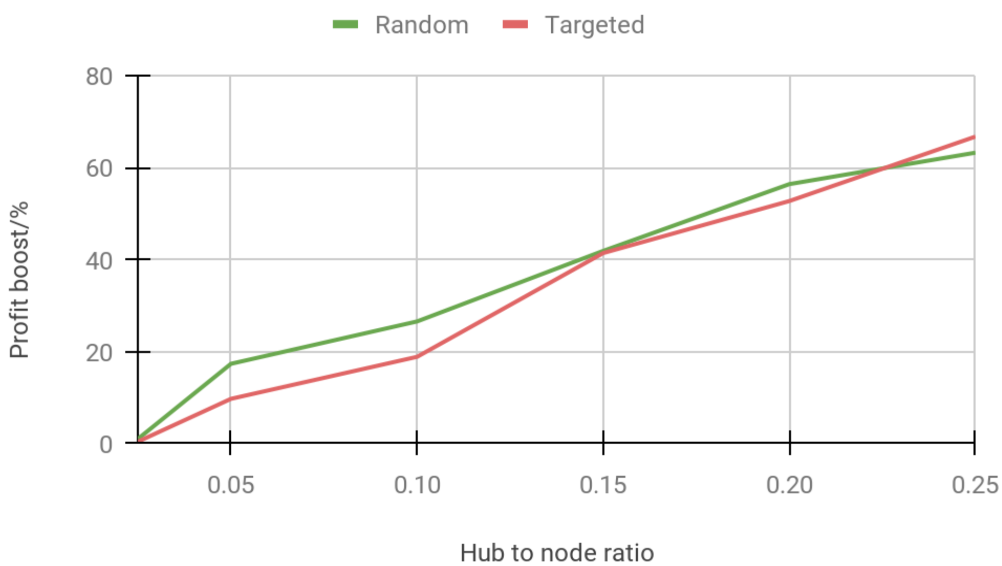

Figure 5: Graph Comparing Different Perturbation Mechanisms with Near-Ideal Parameter Values

From Figure 6, SAILS had the highest mean profit increase, followed by ILS and local search. This shows that our additions to local search were successful in increasing its effectiveness. In addition, Local Search, ILS and SAILS had very low mean coefficients of variation to 3 significant figures of 0.120, 0.106 and 0.106, indicating high stability of the algorithms. However, the low improvement of SAILS over ILS of only 1.13% indicates that there is much room for improvement of SAILS. This still demonstrates SAILS has highest theoretical potential.

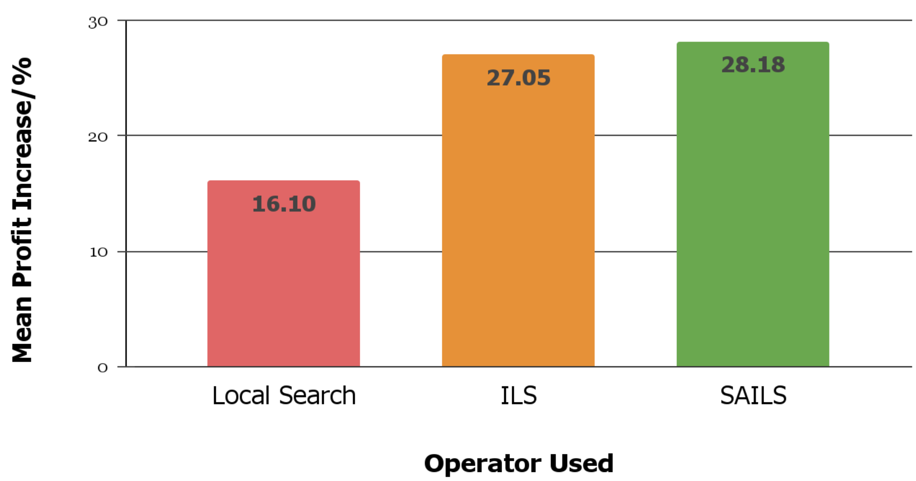

Figure 6: Graph Showing Effect of Operators Used On Mean Profit Increase (n=5)

Figure 7 shows that in general, increasing the time limit caused the mean profit increase to rise, and for each time limit, SAILS displayed the largest mean profit increase, followed by ILS and local search. Hence, SAILS is considered most suitable for use in short time limits. As the time limit increased from 5s to 50s, local search displayed the least increase of 1.59%, as opposed to 7.49% for ILS and 9.11% for SAILS, the greatest increase. This is likely because local search lacks the ability to make large changes which could cause a sudden increase in profit, showing that perturbation and SA increase efficiency and effectiveness. Lastly, SAIL's 50s value of 25.68% is only 2.50% lower than the 20-minute value of 28.18%, and the 5s value of 16.57% is already 58.8% that of 28.18%, showing that SAILS can reach near-optimal results in a very short period of time. This demonstrates practicality, as SAILS can swiftly change routes in responses to new deliveries and do so effectively.

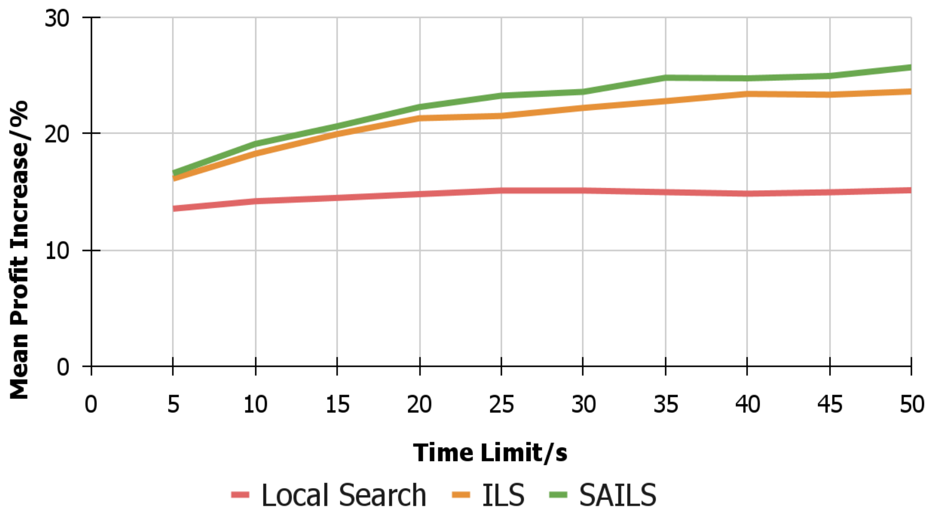

Figure 7: Graph Showing Effect of Time Limit On Mean Profit Increase (n=10)

Figure 8 produces multiple conclusions able to indicate which types of areas SAILS would be more useful for. All algorithms demonstrate increasing effectiveness with higher hub to node ratio, likely due to higher complexity of the system which makes the Greedy Algorithm less effective. With higher vehicle number, effectiveness decreases as the Greedy Algorithm is able to serve more nodes, leaving less work for SAILS. A similar effect occurs with vehicle speed. Maximum distance of node from hub has little, if not a negative effect on SAILS performance if considering two kilometres as an outlier. Yet, effectiveness of local search increases, showing that increase in maximum distance of node from hub reduces utility of Perturbation and SA. This is likely due to the longer average distance of each edge, which decreases the permutations of possible routes, leaving less space for Perturbation and SA. It is concluded that SAILS works better in a higher proportion of shops, but worse with longer distance of nodes from hubs, which implies lower population density.

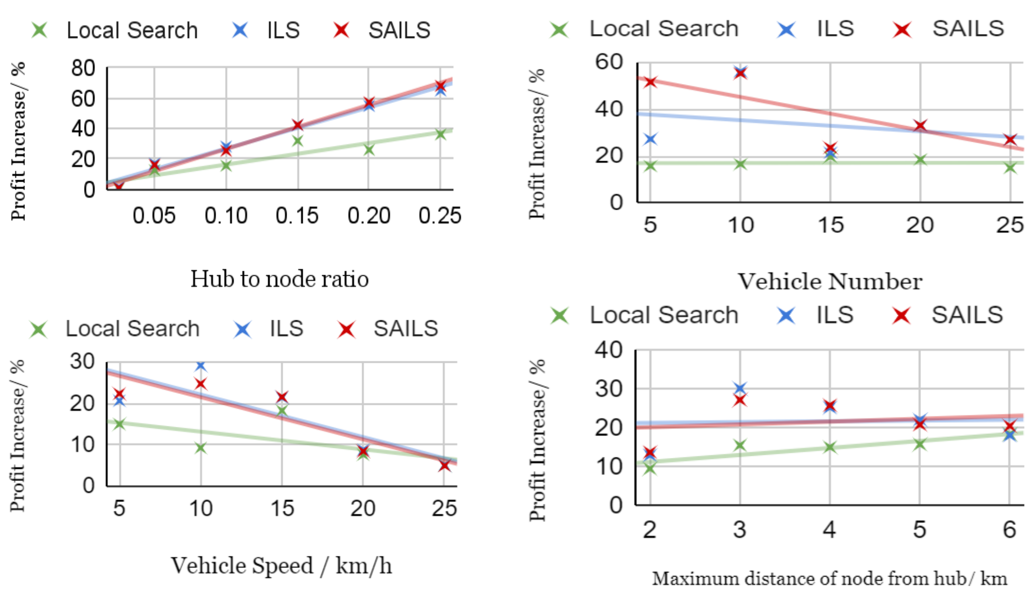

Figure 8: Effect of Variables on Profit Increase

5.2 Proposed Improvements

The limitations of our investigation include assuming that capacity of the vehicles is unlimited, and that the food does not have a specific time window for delivery. The latter limitation can be addressed by using vehicles with short time budgets so that the food is not carried by the vehicle for extended periods of time. An extension of this project could be to implement the capacity and time window limitations into our algorithm to allow for more realistic simulation of food delivery, turning the problem into the Open Capacitated Team Orienteering Problem with Time Windows (Open CTOPTW). Another extension could be to obtain real data from food delivery companies to create more realistic instances, improving accuracy and allowing us to better evaluate practical applicability of SAILS.

Additionally, more investigations can be conducted to enhance our algorithms:

a) Parameter tuning can be made more precise with techniques like ParamILS (Frank n.d.). Another possibility is to relate ideal parameter values to conditions of each instance instead of a one-size-fits-all solution, which would better suit the wide variations in instance variables found across different cities and neighbourhoods.
b) More operators can be implemented. One possible reason for the failure of targeted perturbation was the repeated removal and insertion of the same chunks. To combine the strengths of random and targeted approaches, an algorithm selecting poor chunks with a high chance without 100% certainty may be used (Baker 1987).
c) SA can be improved. For instance, to take into account the possibility of being trapped in deep local optima as the temperature becomes low, reheating to increase temperature after a certain number of iterations has passed may improve performance (Franzin and Stützle 2019).
d) Modifications can be made to instance variables to extend this SAILS approach to other delivery routing problems, such as drone delivery, currently piloted by Amazon, or truck delivery, to solve issues like the supply chain breakdowns the UK experienced.

5.3 Validation

Our aim to create an algorithm for the routing of food delivery services has been successful, as shown by significant increases in profit. Our SAILS algorithm has been proven to be efficient and effective, reaching near-optimal solutions within seconds. In addition, the website can account for dynamism in demand and instances can change based on user input. However, given the NP-hard nature of this problem, it is difficult to know whether the best possible solution has been achieved.

6. Conclusion

In summary, SAILS was successfully applied to the context of food delivery routing modelled as the Open TOP, demonstrating its potential as an effective and efficient routing algorithm. With further improvement of the algorithms and more consideration of real-world constraints, it could become a viable approach to delivery routing, increasing its efficiency and providing greater convenience for businesses and customers alike. It could also accelerate the growth of food delivery services while relieving the overwork many delivery riders experience. As the first study within our knowledge to apply SAILS to the delivery routing context and do so successfully, we believe we have made a unique contribution.

Acknowledgements

We would like to thank the Singapore Ministry of Education Gifted Education Branch and Singapore Management University for giving us this opportunity under the Science Mentorship Programme. Our gratitude also goes to River Valley High School for the unwavering support for this research project.

References

Aarts, E., Korst, J., and Michiels, W., *Search Methodologies,* 1st edition, Springer, 2005.

Alvarez, A., Munari, P., and Morabito, R., Iterated local search and simulated annealing algorithms for the inventory routing problem, *International Transactions in Operations Research*, Volume 25, pp. 1785 – 1809, 2018

Baertlein, L. COVID-19 delivery surge strains FedEx service, opening doors for UPS, Available: https://finance.yahoo.com/news/covid-19-delivery-surge-strains-194748242.html, July 1 2020.

Baker, J.E., Reducing bias and inefficiency in the selection algorithm, *Proceedings of the Second International Conference on Genetic Algorithms and Their Application*, pp. 14 – 21, Hillsdale, New Jersey, USA, July 28-31, 1987.

Coppola, D., Market size of the global online food delivery sector 2020-2023, Available: https://www.statista.com/statistics/1170631/online-food-delivery-market-size-worldwide

Davari S., Sisakht A.H., and Fazel Zarandi A.H., An empirical comparison of simulated annealing and iterated local search for the hierarchical single allocation hub median location problem, *Transactions on Industrial Engineering*, Volume 22, pp. 1203-1217, 2015.

Frank, H., ParamILS, Available: http://www.cs.ubc.ca/labs/beta/Projects/ParamILS/, n.d.

Franzin, A., and Stützle, T., Revisiting simulated annealing: a component-based analysis, *Computers & Operations Research*, Volume 104, pp. 191 – 206, 2019.

Gendreau, M., Laporte, G., and Semet, F., A tabu search heuristic for the undirected selective travelling salesman problem. *European Journal of Operational Research*, vol. 106(2–3), pp. 539 – 545, 1998.

Gera, M., Nawander, N., Tharwani, N., and Bhatia, P., Operations research in food delivery, *International Journal of Advance Research and Development*, vol. 3, pp. 73-78, 2018.

Glover, F. and Kochenberger, G., *Handbook of Metaheuristics*, 1st edition, Kluwer Academic Publishers, 2003.

Golden, B., Levy, L. and Vohra, R. The orienteering problem, *Naval Research Logistics*, vol. 34, pp. 307-318, 1987.

Gottfredson, M., O'Keeffe, D., and Karna, R., Covid-19 has upended business, which trends will stick? Available: https://www.bain.com/insights/covid-19-has-upended-business-which-trends-will-stick, 2021.

Gunawan, A., Lau, H.C., and Vansteenwegen, P., Orienteering problem: a survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, vol. 255, pp. 315-332, 2016.

Gunawan, A., Lau, H.C., Vansteenwegen, P. and Lu, K., Well-tuned algorithms for the team orienteering problem with time windows. J*ournal of the Operational Research Society*, vol. 68, pp. 861-876, 2017.

Hirschmann, R., Singapore: food-delivery spending COVID-19 impact 2020, Available: https://www.statista.com/statistics/1119998/singapore-food-delivery-services-spending-impact-covid-19/, 2020.

Iskandar, H., As demand for delivery surges, food deliverers are not seeing returns, Available: https://www.straitstimes.com/singapore/health/as-demand-for-delivery-surges-food-deliverers-are-not-seeing-returns, 2020.

Martí, R., Pardalos, P., and Resende, M., *Handbook of Heuristics*, 1st edition, Springer, 2018.

McCourt, J., Route optimization software conquers delivery complexity for food distributors, Available: https://www.paragonrouting.com/en-us/blog/post/route-optimization-software-simplified-delivery-routing-for-food-manufacturer/, n.d.

Rajgopal, R., Principles and applications of operations research, Available: https://www.pitt.edu/~jrclass/or/or-intro.html, n.d.

Quek, E., $7k a month as a food delivery rider? Available: https://www.straitstimes.com/lifestyle/food/7k-a-month-as-a-food-delivery-rider, 2020.

Reyes, D., Erera, A.L., Savelsbergh, M.W., Sahasrabudhe, S., and O'Neil, R.J., The Meal Delivery Routing Problem, Available: http://www.optimization-online.org/DB_FILE/2018/04/6571.pdf, 2018.

Sun, G., Tian, Z., Liu, R., Jing, Y., and Ma, Y., Research on coordination and optimization of order allocation and delivery route planning in take-out system. *Mathematical Problems in Engineering,* vol. 2020, pp. 1-16, 2020.

Soria-Alcaraz J.A., Ochoa G., Espinal A., Sotelo-Figueroa M.A., Ornelas-Rodriguez M., and Rostro-Gonzalez H., A methodology for classifying search operators as intensification or diversification heuristics, Available: https://www.hindawi.com/journals/complexity/2020/2871835

Uber., Delivering using the Uber Eats app, Available: https://www.uber.com/us/en/deliver/basics/making-deliveries/how-to-deliver/, 2014.

Van Lon, R. R. S., Ferrante, E., Turgut, A. E., Wenseleers, T., Vanden Berghe, G., and Holvoet, T., Measures of dynamism and urgency in Logistics. *European Journal of Operational Research*, vol. 253, pp. 614-624, 2016.

Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., and Van Oud- heusden, D., Iterated local search for the team orienteering problem with time windows. *Computers and Operations Research*, vol. 36, pp. 3281-3290, 2009.

Biographies

Tan Beng Yang is currently a student at River Valley High School at the Junior College 1 level studying Biology, Chemistry, Mathematics and Geography. He achieved Gold Awards in the Singapore Junior Biology Olympiad, the International Biomedical Quiz, and the National University of Singapore Geography Challenge. He is a nominated candidate for selection into the national team preparing for the International Geography Olympiad. He was a member of the team who triumphed in winning 2nd Runner Up in the National Science Challenge. His previous research project in Environmental Engineering was a finalist entry in the Singapore Science and Engineering Fair. He is also currently Vice-President of the Science Leaders Academy, a school organisation which plans science competitions for schools across the country and aims to nurture students to be leaders in scientific fields. He was also nominated by his school for the Centre for Excellence in Education Research Science Institute, a summer school programme at the Massachusetts Institute of Technology (USA). The annual event sees the congregation of 80 of the world's most accomplished high school science students involving intensive STEM classes and a full research internship with experienced researchers. His current research interests include Oncology and Genetics. However, he is a multidisciplinary person who aims to challenge himself in different areas, such as by embarking on this project. His current career aspirations are to be either a doctor, teacher or public service officer.

Ng Qi Heng is currently a student at River Valley High School at the Junior College 1 level studying Computer Science, Physics, Mathematics and Geography. He is a recipient of the Young Defence Scientists Programme Scholarship. He has an immense passion in Physics. He earned a Silver Award in the Singapore Physics League and a Merit Award in the Singapore Young Physicists' Tournament. He is also currently Vice-Head of the Programmes Department in the Science Leaders Academy, a school organisation which plans science competitions for schools across the country and aims to nurture students to be leaders in scientific fields. He is also the Vice-President of the Audio-Visual Club, which provides audio and backstage support for school events. His research interests include aerodynamics and algorithm design. He aspires to be an engineer or software developer.

Goh Ziyu is a student at River Valley High School at the Junior College 1 level studying Physics, Chemistry, Mathematics and Economics. He received a Gold Award in the International Biomedical Quiz. He also won a Silver Medal in the Singapore Junior Biology Olympiad and an Honorary Mention Award in the Singapore Mathematics Olympiad (Senior Section). His previous research project in Environmental Engineering was a shortlisted entry in the Singapore Science and Engineering Fair. He is currently a member of the Programmes department of the Science Leaders Academy, a school organisation that plans science competitions for schools across the country and aims to nurture students to be leaders in scientific fields. He is also currently a member of the Mathematics Leaders Academy, which seeks to nurture leaders to develop deep subject mastery in Mathematics. His current research interests include Pathology and Biochemistry, but he is open to research in other domains such as Computer Science and Materials Science. His career choices include a medical officer, teacher or engineer in the defence science and technology sector.

Chow Ban Hoe is Head of Department Science Research and Talent Development at River Valley High School (RVHS). He obtained his Bachelor of Science with Honours (Animal Biology) from the National University of Singapore, and his Postgraduate Diploma in Education (Distinction) from the National Institute of Education, Nanyang Technological University, Singapore. He has been teaching Biology, Chemistry and interdisciplinary project work since 2001. In recognition of his active contribution in supporting student science research, he received the Outstanding Science Educator (Research Mentor) Award at the Singapore Science and Engineering Fair in 2015. His team curriculum project on Secondary School Biology education earned the Dr J. M. Nathan Memorial Prize Award in the NIE Management & Leadership in Schools Programme in 2016. Beyond teaching, he also takes on different roles in driving STEM education amongst students and teachers in his capacities as Chairperson of the RVHS Science Leaders Academy and Chairperson of the West Zone Centre of Excellence for Science and Technology.