Heuristic Path Planning Approach for a Granular-fill Insulation Distributing Robot ⋆

Peter Gsellmann * **Martin Melik-Merkumians** *
Milan Hurban * **Georg Schitter** *

* *Automation and Control Institute (ACIN), TU Wien, 1040 Vienna, Austria (e-mail: {gsellmann, melik-merkumians, hurban, schitter}@acin.tuwien.ac.at)*

Abstract: In this paper, a heuristic path planning approach for the robotic distribution of granular-fill insulation material is presented. The initial coarse manual distribution of the material leads to an uneven surface with areas of excessive or insufficient material. In order to distribute the granular-fill insulation uniformly with a robot, first the worked area is captured as point cloud with an RGB-D camera, and afterwards these irregularities are located via agglomerative hierarchical clustering. Subsequently, their volumes are estimated providing weights for the path calculation. A path planning method, inspired by the usual working method of human construction workers, is developed and applied. In a test scenario, the total path length and the processing sequence are analysed, varying the blade size and the weight of the distance-to-goal parameter. This analysis yields, that the presented path planning algorithm is well suited for the described application, showing the best results with a larger blade size and a quadratic distance-to-goal behavior.

Keywords: Path planning, Heuristics, Robotic manipulators, Vision, Search methods, RGB-D cameras

1. INTRODUCTION

In recent years, building construction has become an important research area in the field of robotics (Balaguer and Abderrahim (2008)). One particular application of interest is the automated distribution of floor insulation material, since this is a physically demanding task. With this motivation, Hurban et al. (2019) modified the manually operated platform (patent of Karl-Heinz Müller (2006)), by means of motorising and controlling the manipulator, with the aim of an automated distribution of granular-fill insulation material. The considered system, shown in Fig. 1, is a SCARA-like robotic arm with a distribution blade as its end-effector tool. This blade is automatically height adjusted by a laser reference. The arm itself is mounted on a portable automatically leveled tripod base. Due to the importance of the system's transportability, it is battery-powered and realised as a lightweight construction.

For the use in floor installation work, the robot is manually set-up at the desired position. Subsequently, the granular-fill insulation is distributed roughly over the area to be processed. The resulting surface of the distributed material shows an uneven character. In order to perform the final smoothing sweep, a sufficient amount of material over the entire working area is required. This raises the question on how to find a suitable path which leads the robot to fill the surface valleys by gathering and accumulating material from areas with a surplus. Additionally, the path is also expected to progress from the back to the front of the area.

Fig. 1. Granular-fill Insulation Distributing Robot with SCARA-like configuration. An actuated arm with an end-effector tool is used to distribute the filling material in order to smooth the to be processed surface.

On a more abstract level, this task describes a routing problem where the irregularities are considered as the to be connected nodes, weighted by their type, size, and distance to the goal. Many exact and heuristic approaches have been developed and analysed in order to solve similar problems, such as the Traveling Salesman Problem (TSP), Open Vehicle Routing Problem (OVRP), and Capacitated Vehicle Routing Problem (CVPR) (Toth and Vigo (1998); Baraglia et al. (2001); Pereira and Tavares (2009)).

⋆ The financial support by the Austrian Research Promotion Agency (FFG) under grant no. 866395, as well as Mixit Dämmstoffe GmbH is gratefully acknowledged.

For applications with a high number of nodes Prins (2004) states, that heuristic solutions typically come out as more flexible and simpler techniques compared to exact solution methods. Representatives for proposed metaheuristic methods with the aim of solving a routing problem are the Ant Colony Optimization (ACO), the Genetic Algorithm (GA), Simulated Annealing (SA), or tabu search, as presented in Lee et al. (2008); Vidal et al. (2013); Yu and Lin (2015); Qiu et al. (2018).

Moreover, several planning approaches with machine learning methods, for instance Q-learning and unsupervised fuzzy clustering, were conducted in Gambardella and Dorigo (1995); Maire and Mladenov (2012); Ewbank et al. (2015).

The contribution of this paper is the path planning concept for a granular-fill insulation distributing robot. In Section 2, the preparatory work for the use of the path planning algorithm is described. First, a level plane has to be detected in order to find the position of the irregularities and afterwards their volume is estimated. In Section 3, the algorithm for the path planning and its implementation is explained in detail. Simulation results are presented in Section 4. The conclusion and outlook are given in Section 5.

2. PRELIMINARIES

After the granular-fill insulation material is roughly placed in the target area, heaps and valleys have to be detected. The whole scenery, depicted in Fig. 2, is captured with a *Microsoft Kinect v2* RGB-Depth/Distance (RGB-D) camera. With help of the RGB picture, the reference height, provided by a leveling laser, is obtained. Additionally, the volumes of the irregularities are estimated.

2.1 Level Plane Detection

In order to calculate the level plane, an RGB image of the camera is used. The initial task is to detect the red laser line. This is achieved by first extracting the red component of the RGB picture. Areas with a low red value are neglected. Subsequently, the noise of the image is filtered out with a 3x3 median filter and connected pixel clusters with 30 objects or fewer are excluded. With the remaining image, the parameters of the laser line are calculated with Random Sample Consensus (RANSAC) by Fischler and Bolles (1981). Afterwards, the subsequent step is to extrapolate from the detected laser line to the level plane. Since the RGB and the depth sensor of the 3D camera are two different devices at different positions, a function for mapping, as stated in Terven and Córdova-Esparza (2016), is necessary. Thus, the transformation of the detected line from the RGB to the 3D image is enabled. Assuming that the camera is oriented parallel to the level plane, the normal vector of the latter can be calculated by geometric considerations (cf. Fig. 2). The 3D vector v_L of the laser projection on the background is calculated according to Eq. 1a. Here, v_1 and v_2 are defined as vectors from the camera position to two points on the laser line. The vector v_C defines the direction of the camera position to the center of the image. Since v_L and v_C are direction vectors of the desired level plane, their cross product results in the planes normal vector n (cf. Eq. 1b).

$$v_L = v_1 - v_2 \tag{1a}$$

$$n = v_C \times v_L \tag{1b}$$

Afterwards, the calculated plane can be placed virtually into the scenery of the area to be processed in order highlight the irregularities, exemplary shown in Fig. 3. Here the areas with too much material (brown) protrude over the plane, while areas with too little material (green) remain below it.

2.2 Determination of the Position of the Irregularities

For the determination of the irregularities in the surface, the recorded point cloud is split into two parts: a set above and a set below the level plane.

Subsequently, an agglomerative hierarchical clustering algorithm, as presented in Gan et al. (2007), is used to dissect the point cloud and to find the positions of the heaps and valleys. Clusters exceeding a certain number of points are partitioned iteratively, until the restriction of maximum cluster size is met. Also small irregularities close to each other are combined if they have the same polarity. In this context, polarity defines the type of irregularity: either areas with excessive material or areas with insufficient material. By defining the clusters for the heaps and valleys, their volumes can be estimated by simply counting the number of points assigned to them.

3. PATH PLANNING

The requirements of the present application are only partially consonant with the classic Vehicle Routing Problem (VRP). Firstly, only one path from a fixed start node to a fixed end node is required. Secondly, the nodes are not equivalent for the material distribution problem, since they represent different sized areas with too much or too

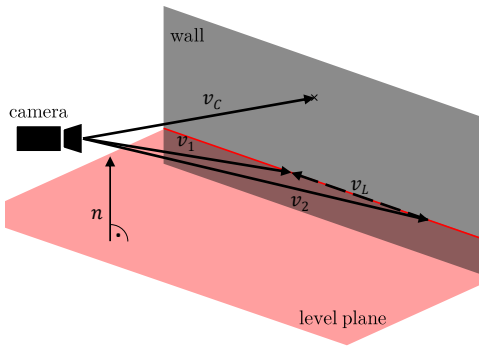


Fig. 2. By forming the difference of the vectors v_1 and v_2, which are vectors from the camera to two points on the laser line, the 3D direction vector of the laser projection v_L can be obtained. Assuming the camera orientation is parallel to the desired level plane, the normal vector of the aforementioned is calculated according to Eq. 1b.

Fig. 3. A laser level is used to highlight the desired floor height (red). Knowing the orientation of the camera, an artificial plane is placed in the point cloud to point out the areas with a too high (brown) or insufficient (green) amount of material.

little material, respectively. This requires a systematic path planning algorithm, which incorporates the aforementioned needs.

The basic concept of the path planning algorithm strives to mimic the working principle of construction workers. They start at the back of the area to be processed in order not to create unevenness of an already smoothed surface. The workers take material from heaps and distribute it with a shovel to areas with too little insulation material.

This approach is mimicked by the algorithm, for planning the necessary path of the robot manipulator for an even distribution.

As a first approach, an alternating behavior between low and high positions is considered. However, as seen in the illustrative example shown in Fig. 4, this method can lead into the problem of not filling holes sufficiently. This can be explained by the fact, that an alternating behavior only considers the polarity of the irregularities. For that reason, the volume of the heaps and valleys, and the volumetric capacity of the blade has to be considered for an efficient path planning algorithm.

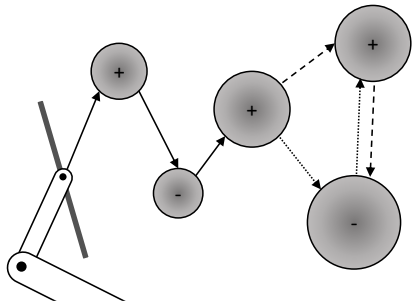


Fig. 4. Motion Planning for the Distribution: By just alternating between lows (-) and highs (+) (solid and dotted line), the last low in this example is not filled sufficiently. If the motion planning allows the accumulation of material, the second approach (solid and dashed line) shows better behavior.

3.1 Planning Algorithm

This section presents the developed path planning algorithm in detail. In order to better exemplify the path planning method, a pseudo code is presented in Algorithm 1. The start and the goal position, the position of irregularities and their weights are handed over as input variables, since they are obtained in a higher planning level. Initially, in Line 1 to 5, initialisations are performed. The start position is provided as the input variable p and is added to the path *solution* as a first entry. Then the current and the maximum number of points for the volumes of the blade n_V and $n_{V_{max}}$ are assigned. The target indicates what kind of irregularity is headed for next, whereas a high is defined as an area with a surplus of material, while a low indicates an area with insufficient material. In this application, the algorithm aims first for a high, since it needs to accumulate material in the beginning, which corresponds to the construction worker's intuitive initial move to a material heap.

The Euclidean distance of every center position of the irregularities to the goal is stored in w_n (cf. Eq. 2).

Algorithm 1 Planning algorithm

Input: start position p**, goal position** *goal*, **position of irregularities** *centers*, **volume weights** n_{V_c}
Output: path *solution*

1: Initialise $n_{V_{max}}$
2: Initialise n_V to 0
3: Initialise target to a high
4: $w_n \leftarrow calculateWeights(centers, goal)$
5: $solution \leftarrow p$
6: **while** list of centers is not emtpy **do**
7: $[d_C, c_n] \leftarrow nearestCenter(p, centers, target, w_n)$
8: **if** the target is a low and $n_V < n_{V_{max}}$ **then**
9: $[d_L, c_L] \leftarrow checkNextLow(p, centers, w_n)$
10: **if** $d_L < d_C$ **then**
11: $d_C \leftarrow d_L$
12: $c_n \leftarrow c_L$
13: **end if**
14: **else if** the target is a high and $n_V > 0$ **then**
15: $[d_H, c_H] \leftarrow checkNextHigh(p, centers, w_n)$
16: **if** $d_H < d_C$ **then**
17: $d_C \leftarrow d_H$
18: $c_n \leftarrow c_H$
19: **end if**
20: **end if**
21: **if** no c_n is found **then**
22: $[x, y] \leftarrow getLinearPath(p, goal)$
23: $solution \leftarrow [solution; [x, y]]$
24: **break**
25: **end if**
26: $[x, y] \leftarrow getLinearPath(p, c_n)$
27: $solution \leftarrow [solution; [x, y]]$
28: Recalculate n_V
29: Remove current center from centers
30: Remove current volume weight from volume list
31: Remove current distance weight from weight list
32: Invert target
33: $p \leftarrow c_n$
34: **end while**

$$w_n = \|c_n - goal\|_2 \tag{2}$$

Afterwards, in Line 6, the procedure is entering a while loop, which is terminated if either no more appropriate centers are found or the list of centers is empty. At first, the nearest centers are checked via the *nearestCenter* procedure, which requires the current position p, the centers of the irregularities *centers*, the target and the distance weights to the goal w_n as parameters. The return values of the aforementioned function is the nearest irregularity c_n of the polarity which is defined with the target and its distance to the current position d_C (for more detail see Section 3.2).

From Line 8 to 20, the algorithm checks if an irregularity with the same polarity as the current center is more suitable in terms of distance, weight, and volume. If the current position is at a high, and therefore the next regular center would be a low, it may occur, that a center with a positive polarity is closer and/or has a lower weight. If now the capacity of the blade is not stretched and the volume of the suitable positive center would not exceed it, the latter turns up as a superior target candidate. Analogously, if the current position is at a low, and the distribution blade has still enough material for a valley with more favorable properties, it is chosen over the regular candidate. This behavior corresponds to the presumable considerations of the construction worker: If the trowel is able to hold another close material heap, the worker aims for it. On the contrary, if the trowel is full, the worker aims for the nearest area with insufficient material. In order to realise the aforementioned functionality, the procedures *checkNextLow* and *checkNextHigh* are used. In Lines 26 and 27, the path from current position p to the chosen target c_n is calculated and added to *solution*. Here, the procedure *getLinearPath* delivers a linear connection between these two points. Afterwards, in Line 28 the blade volume is recalculated based on the current center. At the end of the while loop, the current position is set to the position of chosen target center. The latter is then also removed from the list of centers. Additionally, the corresponding volume and weight entry is deleted. As the last statement, the polarity is inverted. The procedure ends, if *nearestCenter* in Line 7 returns $c_n = [\,]$, which means no suitable irregularity is found. Hence, the condition in Line 21 is fulfilled and the algorithm considers the goal as the last target. Finally, the path from the current position to the goal is added to *solution*.

3.2 Nearest Center Procedure

The procedure for the search of the next center takes the current position p, the irregularities *centers*, the target, and the distance weights w_n as parameters. For every center, a weight d_n is calculated according to Eq. 3.

$$d_n = \|p - c_n\|_2 + w_n^k \tag{3}$$

As can be seen, not only the distance from the current position to the center, but additionally the weight of the distance to the goal is considered. This weight can be raised to the power of k, which increases the effect of this distance. Since a construction worker is processing the area from the rear side to the front with the intention to not enter the processed region again, the presented procedure supports this desired behavior.

Table 1. Number of points for calculated centers of the exemplary test case. Since the total volume of the areas with excessive material is higher than the total volume of the lows, the area contains enough material for the distribution.

	Highs	**Lows**
Total Number	22	8
Maximum Volume	160	89
Minimum Volume	2	3
Total Volume	646	204

4. SIMULATION RESULTS

In this section, simulation results of the developed heuristic path planning approach are presented. For this purpose, an actual construction site scenario is used as a test case, where initially the material is distributed coarsely within the desired area. The recorded point cloud is shown in Fig. 5.

With the preliminary steps finished, the positions and volumes of 30 irregularities are obtained (cf. Table 1). As can be seen, the overall volume of areas with excessive material is higher than the volume of the valleys, which is why no more material needs to be added to the surface and all valleys can be filled.

The two parameters, blade volume and distance weight coefficient, are influencing the selected path, which have to be analysed. Considering this quasi-static approach, the path length and the processing sequence appear as reliably measurable criteria.

4.1 Analysis on the Variation of the Maximum Blade Volume

As the size of the blade, and therefore the maximum volume of material stored in front of it, changes the path planning substantially, an according analysis is conducted.

The first experiment, depicted in Fig. 6(a), is performed with a maximum blade volume of 180 points. Considering that the biggest center has a volume of 160 points, the algorithm needs to adapt its path planning accordingly. As a consequence, the robot manipulator has to alternate more often between the irregularities of different polarity

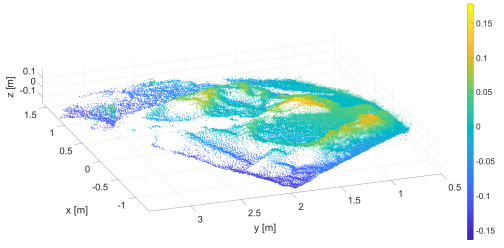
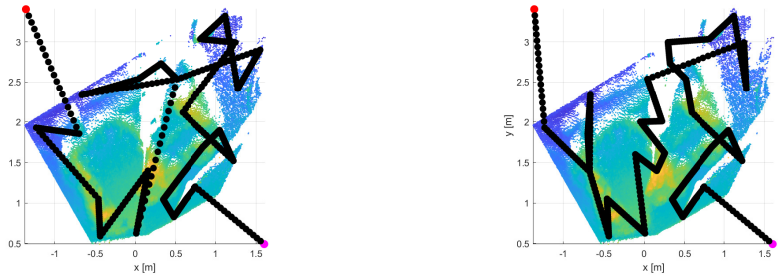


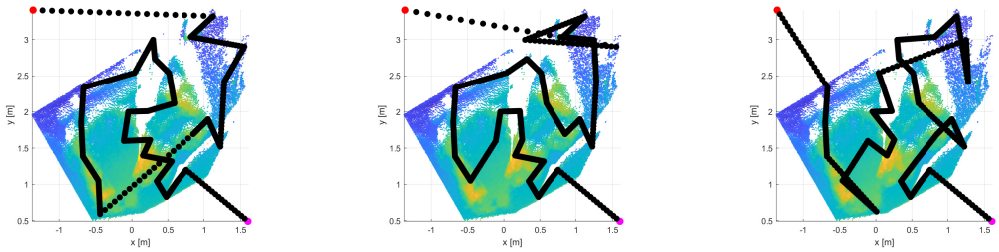
Fig. 5. Point cloud of the area to be processed as the initial situation, where the granular-fill material is roughly distributed over the area.



(a) Distribution blade volume 180 and quadratic consideration of distance to goal

(b) Distribution blade volume 475 and quadratic consideration of distance to goal

Fig. 6. Distribution path of robot with different maximum blade volume and quadratic consideration of distance weighting, calculated with presented heuristic for an actual exemplary floor configuration, recorded as a point cloud via a 3D camera. The computed path (black) begins at the starting point (magenta) and continues until the end point (red).



(a) Distribution blade volume 750 without consideration of distance to goal

(b) Distribution blade volume 750 and linear consideration of distance to goal

(c) Distribution blade volume 750 and quadratic consideration of distance to goal

Fig. 7. Distribution path of robot with constant blade volume and with linear, with quadratic or without consideration of the distance to the goal, calculated with presented heuristic for an actual exemplary floor configuration, recorded as a point cloud via a 3D camera. The computed path (black) begins at the starting point (magenta) and continues until the end point (red).

which leads in a longer path. For this blade the complete path results in a length of 18.89 m.

The second experiment is conducted with maximum blade volume of 475 points, which represents the size of the currently used blade. The simulation result is presented in Fig. 6(b). Compared to the first experiment, the path seems more reasonable considering a shorter total path length of 17.86 m. In general, the choice of the blade size will lead to a trade-off between flexibility and weight of the manipulator versus the maximum possible material load.

4.2 Analysis on the Distance Weighting

A second analysis is conducted based on the distance weighting of the irregularities. Since it is desired to work the surface from the back to the front, the distance from each center to the goal point is used as a weight (cf. Eq. 3). It is of interest to know which k yields the best results for the path planning. For this reason, the cases $k = \{0, 1, 2\}$ are analysed.

For the analysis of the calculated paths shown in Fig. 7, a constant maximum blade volume of 750 points is chosen. The simulation result for the example, without considering the distance weighting to the goal, by setting $k = 0$, is shown in Fig. 7(a). With a total length of 15.77 m, the planning algorithm without the distance weighting manages to generate an appropriate path. However, the sequence is not progressing from the rear end to the front anymore, which is crucial for the considered industry.

For the second experiment with $k = 1$, which indicates a linear dependency of the weight to the distance, the results in Fig. 7(b) could be obtained. A path, reaching all centers and taking the back-to-front restriction into account, with a total length of 16.29 m is generated.

The last simulation covers the case for $k = 2$, which stands for a quadratic dependency of the distance weight. The calculated path with a total length of 15.33 m can be seen in Fig. 7(c). Considering that the length is 3% less than in the simulation with $k = 0$ or 6% less than in the simulation with $k = 1$, and the processing path does comply with the requirement of working from the back to the front, the method with $k = 2$ shows the best results in this analysis.

An overview of the lengths for the obtained paths of both experiments can be seen in Table 2. The parameters $k = \{0, 1, 2\}$ and $n_{V_{max}} = \{180, 475, 750\}$ are taken into

Post-print version of the article: P. Gsellmann, M. Melik-Merkumians, M. Hurban, and G. Schitter, "Heuristic Path Planning Approach for a Granular-fill Insulation Distributing Robot," *21st IFAC World Congress*, 2020.

Table 2. Length of calculated paths for the analysis on the variation of the maximum blade volume $n_{V_{max}}$ and the degree of distance dependency to the goal k.

	$\mathbf{n_{V_{max}}}$		
k	180	475	750
0	19.15 m	18.07 m	15.77 m
1	19.87 m	25.63 m	16.29 m
2	18.89 m	17.86 m	15.33 m

account. As can be seen, the longest path is achieved in the simulation with a blade size of 475 points and linear dependency from the centers to the goal. On the contrary, the quadratic distance dependency from the centers to the goal leads to the shortest necessary path for all analysed blade sizes.

In summary, the proposed path planning approach is well suited for the intended distribution task, delivering the shortest sufficient routes in the case of a larger sized blade and a quadratic distance-to-goal dependency.

5. CONCLUSION

In this paper, a heuristic path planning approach for a granular-fill insulation distribution robot is successfully developed and analysed. With the aim of detecting the irregularities of the surface, an RGB-D camera is used. Preliminary steps comprise the detection of the 3D level plane, and the calculation of the position of the centers and the volume estimates of the irregularities. With these steps done, the developed path planning algorithm, based on the manual working principle of construction workers, can be used to obtain a path for the distribution of the material.

The analysis on the variation of parameters for this heuristic approach yields, that the best results are achieved with a larger blade size and a quadratic distance dependency to the goal.

Ongoing work will focus on the analysis of different sensor and camera concepts. Furthermore, a method to estimate the overall material volume has to be developed. In order to fill all surface valleys, this gives the user the information, whether the area contains a sufficient amount of material. Also, further research concerning the path planning algorithm has to take the velocity and dynamics of the robot into account. Additionally, the continuation of the distribution of the material after relocating the robot has to be analysed.

REFERENCES

Balaguer, C. and Abderrahim, M. (2008). Trends in Robotics and Automation in Construction. In *Robotics and Automation in Construction*. InTech.

Baraglia, R., Hidalgo, J., and Perego, R. (2001). A hybrid heuristic for the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 5(6), 613–622. doi:10.1109/4235.974843.

Ewbank, H., Wanke, P., and Hadi-Vencheh, A. (2015). An unsupervised fuzzy clustering approach to the capacitated vehicle routing problem. *Neural Computing and Applications*, 27(4), 857–867. doi:10.1007/s00521-015-1901-4.

Fischler, M.A. and Bolles, R.C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395. doi:10.1145/358669.358692.

Gambardella, L.M. and Dorigo, M. (1995). Ant-Q: A Reinforcement Learning approach to the traveling salesman problem. In *Machine Learning Proceedings 1995*, 252–260. Elsevier. doi:10.1016/b978-1-55860-377-6.50039-6.

Gan, G., Ma, C., and Wu, J. (2007). *Data Clustering: Theory, Algorithms, and Applications*. Society for Industrial and Applied Mathematics.

Hurban, M., Melik-Merkumians, M., Steinegger, T., Bibl, M., Gsellmann, P., and Schitter, G. (2019). Automated Tripod Leveling and Parameter Estimation for a Granular-fill Insulation Distributing Robot. In *Proceedings of the Joint Conference 8th IFAC Symposium on Mechatronic Systems, and 11th IFAC Symposium on Nonlinear Control Systems*.

Karl-Heinz Müller (2006). Loose dry material leveling apparatus, e.g. for expanded concrete granules laid on the ground, has a horizontal pull-out unit in a carrier at a holder on a telescopic arm.

Lee, C.Y., Lee, Z.J., Lin, S.W., and Ying, K.C. (2008). An enhanced ant colony optimization (EACO) applied to capacitated vehicle routing problem. *Applied Intelligence*, 32(1), 88–95. doi:10.1007/s10489-008-0136-9.

Maire, B.F.J.L. and Mladenov, V.M. (2012). Comparison of neural networks for solving the travelling salesman problem. In *11th Symposium on Neural Network Applications in Electrical Engineering*. IEEE. doi:10.1109/neurel.2012.6419953.

Pereira, F.B. and Tavares, J. (eds.) (2009). *Bio-inspired Algorithms for the Vehicle Routing Problem*. Springer Berlin Heidelberg. doi:10.1007/978-3-540-85152-3.

Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12), 1985–2002. doi:10.1016/s0305-0548(03)00158-8.

Qiu, M., Fu, Z., Eglese, R., and Tang, Q. (2018). A Tabu Search algorithm for the vehicle routing problem with discrete split deliveries and pickups. *Computers & Operations Research*, 100, 102–116. doi:10.1016/j.cor.2018.07.021.

Terven, J.R. and Córdova-Esparza, D.M. (2016). Kin2. A Kinect 2 toolbox for MATLAB. *Science of Computer Programming*, 130, 97–106. doi:10.1016/j.scico.2016.05.009.

Toth, P. and Vigo, D. (1998). *Exact Solution of the Vehicle Routing Problem*, 1–31. Springer US, Boston, MA. doi:10.1007/978-1-4615-5755-5_1.

Vidal, T., Crainic, T.G., Gendreau, M., and Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, 40(1), 475–489. doi:10.1016/j.cor.2012.07.018.

Yu, V.F. and Lin, S.Y. (2015). A simulated annealing heuristic for the open location-routing problem. *Computers & Operations Research*, 62, 184–196. doi:10.1016/j.cor.2014.10.009.