Order picking: a comparison of heuristic and meta-heuristic approaches

Bertolini M.*, Melloni R.*, Neroni M.**

** Dipartimento di Ingegneria "Enzo Ferrari"., Università degli Studi di Modena e Reggio Emilia, via P. Vivarelli, 10 - 41125 Modena – Italy (massimo.bertolini@unimore.it; riccardo.melloni@unimore.it)*

*** Dipartimento di Ingegneria e Architettura, Università degli Studi di Parma, Via G.P. Usberti, 181/A – 43124 Parma – Italia (mattia.neroni@unipr.it)*

Abstract: This paper investigates the impact of several routing strategies on minimization of the travel-time for pickers in a manual warehouse. Many solutions have been proposed in literature over the last years, however, just a few authors really compare them on equal terms in order to carry out a proper comparison. In this work, three well-known heuristic strategies (i.e. S-Shape, Largest Gap, and Combined) are firstly compared to each other and then with two meta-heuristic algorithms owning to swarm intelligence family (i.e. Ant Colony Optimization and Particle Swarm Optimization). Firstly, an empirical study has been made to find out the best setting for meta-heuristics' parameters. Then, a discrete event simulation model has been developed by using both Python© and Cython© programming languages, and the analyzed strategies have been compared under several storage assignment policies.

Keywords: order picking, heuristic, meta-heuristic, travel sales problem, logistic.

1. Introduction

Warehousing is an aspect of paramount importance in supply chain and logistics. It consists in storage for raw materials or finished goods, in order to provide a good synchronization of industrial processes. Nowadays, it is uniformly achieved that a well-designed warehousing policy leads to benefits and positive impacts, within the company and across the entire supply chain (Zhang and Lai, 2006). Thanks to the inventory, the risk of delay in shipments and production is considerably reduced, even if the demand is subject to big changes (Penteado et al. 2016). Furthermore, warehousing offers a wide range of economic benefits and increases the value of goods, since the products are available at the right place in the right time (Penteado et al. 2016). Over the last years, thanks to innovative automation technologies, new efficient automated solutions spread and found application in several industrial fields (Bertolini et al., 2019). In particular, concerning logistic, more and more manual warehouses were replaced by partially and fully automated solutions, namely automated storage and retrieval systems (AS/RS). The new AS/RSs are able to fulfil more shipments per day, and their storage capacity is much higher if compared to manual warehouses. They represent a great success, and this transition to automation is now well-known as "automation logistic". However, even if the fully automated solutions are usually more efficient, their implementation is strictly limited to contexts in which unit loads are standardized, quantities are big, and the variety of retrieved items is small (Janssen et al. 2019). For this reason, in contexts of small-items-storage and spare-parts-handling, manual warehouses still represent the standard solution. In most of distribution centres, where operations usually consist in decomposition of pallets, short period storage, re-composition and shrink-wrapping of new pallets, classic picker-to-parts strategy is still the most widespread. Since, the pickers spend the most of their working time moving from a warehouse location to another, effective picking strategies may reduce the travel time and increase the productivity. Each warehouse is different, and the picking strategy must be chosen based on layout, items, number of resources, and equipment that the workers are utilizing. Among the most known picking strategies are batch picking, zone picking, and wave picking. Many other strategies, such as vision picking and voice picking, also found application thanks to innovative electronic solutions. However, all these strategies require particular conditions or equipment to be efficiently implemented. For instance, batch picking and wave picking need the possibility to sort the items after retrieving, zone picking is efficient in case of very diverse inventory, vision picking and voice picking need an headset and a mobile computer attachment. For these reasons, classic order picking, even though it guarantees less chance of improvement, is often the most sensible choice. Since, in case of order picking there is no batches composition problem, the main aspect to focus on is the routing of pickers. The scientific literature has been studying for long time the routing strategies to reduce the travel time for pickers under order picking conditions, although, each time a new approach is proposed, it is often compared on equal terms with just one or two more solutions (Hosam et al. 2015). Moreover, to the authors' best knowledge, only a few publications exhaustively study the impact of different layouts on the proposed approach. Because of this, during the design phase of a manual warehouse, is hard to understand which is the most feasible approach. In this paper, the aim of the authors is to partially fill this gap. Focusing on order picking, three heuristic routing strategies (i.e. S-Shape, Largest Gap, and Combined) are firstly compared each other, and then compared to a meta-heuristic algorithm owning to swarm intelligence family (i.e. Ant Colony Optimization). The choice of the ant colony is due to the fact that some publications sponsor it as the most effective in these contexts. Finally, the authors take the opportunity to present their own solution by introducing a re-adapted Particle Swarm Optimization, which is lastly compared with previously introduced approaches. The Particle Swarm Optimization (PSO) is a relatively new metaheuristic, which, up to now, has always provided great benefits when applied to different contexts such as Neural Networks training (Suresh et al., 2015) and Bayesian Networks (Aouay et al., 2013). Its application to industry, logistics, and other contexts where a discrete version of the algorithm is needed is still premature, however, in the era of modern computing, the PSO is promisingly more suitable than other metaheuristics for both parallelization techniques: multiprocessing and multi-threading (Malakhov et al., 2018). All the comparisons of the analysed algorithms are made using the same layout, and observing the behaviour of the analysed approaches when the complexity of the problem and the storage assignment policy change. The reminder of this paper is structured as follows. In section 2 the heuristic strategies are briefly described. In section 3 the Ant Colony Optimization (ACO) and the proposed version of Particle Swarm Optimization (PSO) are described. An empirical setting of the parameters of ACO and PSO in described in section 4. In section 5, the layout used is introduced and the proposed approaches are compared each other. Finally, considerations and futures developments are presented in section 6.

2. Heuristic approaches

2.1. S-Shape

The S-Shape strategy (Marchet, 1994) is also known as 'traversal'. It defines a route in which the aisles that are to be visited, are totally traversed. Conversely, aisles where nothing has to be picked are skipped. The advantage of this strategy consists in its simplicity and easy implementation. In case of multiple blocks and cross-aisles, after traversing an aisle, to decide which aisle to run, only adjacent blocks are considered.

2.2. Largest Gap

In the Largest Gap (LG) strategy, firstly introduced by Hall (1993), every time the picker enters an aisle, the distance between the current picking position and the next one is estimated, if it is bigger than the distance between the current position and the beginning of the aisle, the picker go back to the beginning of the aisle, otherwise he/she goes to the next picking position. In case of multiple blocks and cross-aisles, as for the S-Shape strategy, every time a new picking point is to be defined, only adjacent blocks are considered.

2.3. Combined

The Combined strategy (Roodbergen and De Koster, 2001) introduces a sort of try-and-error evaluation to define the best route. Every time all picking position in an aisle have been visited, the alternatives to go to the rear end of the aisle and to return to the front end are compared with each other. The solution resulting in the shortest path is chosen. As for the strategies described above, in case of multiple blocks, only the adjacent ones are considered.

3. Meta-heuristic approaches

3.1. Ant Colony optimization (ACO)

The ACO is a meta-heuristic optimization technique inspired by ants behaviour. When an ant must choice a route instead of the other, he/she looks at the quantity of pheromone left by other members of the colony. An higher level of pheromone means a better route, usually because shorter if compared to the others. This curious behaviour inspired the creation of a probabilistic technique of operational research for solving computational problems, which can be formalised with a graph. The first version was proposed by Dorigo et al. (1996), and it was originally called Ant System. Then, over the years, several improvements and adjustments to different contexts were proposed (Bell and McMullen, 2004), and the ACO was rearranged to work with discrete problems and is now note to be one of the best performing algorithms for routing and Travel Sales Problem (TSP). The ACO execution consists of loops. At each loop t, a new ant is generated and the algorithm takes into account the set of n picking positions $(i = 1,\dots,n)$ to be visited. The edge connecting two picking positions i and j, where $i \neq j$, can be denoted by tuple (i, j), and its length is $d_{i,j}$. Hence, the cost of a solution D may be calculated as $\sum_{i=1}^{n-1} d_{i,i+1}$.

Each ant provides a new solution by building it element by element. The new solution is then compared to the best solution found so far and, if its cost is lower than the best solution's cost, it is made the new best solution. Every time a new best solution is found, the pheromone on each edge is updated. Given $\tau_{i,j}$ the pheromone on edge (i, j), it is updated according to eq.(1), where ρ and Q are parameters of the algorithm.

$$\tau_{i,j} = \begin{cases} \rho\tau_{i,j} + \dfrac{Q}{d_{i,j}}, if\ (i,j) \in best\ solution \\ \rho\tau_{i,j}, \qquad otherwise \end{cases} \tag{1}$$

While the ant is building a new solution, given i the last element of the current incomplete solution and I the set of picking positions not visited yet, the next position is selected by using a roulette wheel (Shtovba, 2005), where the probability to move to picking position j, namely $p_{i,j}$, is calculated as in eq.(2).

$$p_{i,j} = \begin{cases} \dfrac{(\tau_{i,j})^{\alpha}\left(\dfrac{1}{d_{i,j}}\right)^{\beta}}{\sum_{k\in I}(\tau_{i,k})^{\alpha}\left(\dfrac{1}{d_{i,k}}\right)^{\beta}}, if\ j \in I \\ 0, otherwise \end{cases} \tag{2}$$

3.2. Particle Swarm Optimization (PSO)

The PSO is a meta-heuristic optimization technique inspired by bird flocks behaviour. Each bird is represented by an element called particle, and, at every iteration of the algorithm, he/she explores a new solution in its neighbourhood according to his/her own decisions and other particles decisions. The first contribution to PSO is attributed to Kennedy and Eberhart (1995), and it was firstly intended as a simulation of social behaviours and not as an optimization technique. Contrary to ACO, there are not many scientific contributions proposing a discrete version of PSO, and most of them propose a binary formalization. In the original version, given t the current iteration, each particle has a current position or current solution called *current(t)*, and a personal best called *pbest(t)*, which represents the best solution explored by the particle. Moreover, each particle knows the global best found so far, which is usually called *gbest(t)* and represents the best solution explored by the whole swarm. The solution explored by each particle in iteration t is defined by its current position *current(t)* and its current speed $v(t)$, where $v(t+1)$ may be calculated as in eq.(3).

$$v(t+1) = v(t) \cdot w + rnd \cdot C_1(pbest(t) - current(t)) + rnd \cdot C_2(gbest(t) - current(t)) \quad (3)$$

Where w, C_1 and C_2 are parameters of the algorithm, and *rnd* is a random number in range *[0,1]*. Clearly, this behaviour is not possible in a discrete context, because of this, the first discrete PSO algorithms proposed (Shi et al. 2007), are even more popular than the original one. In these publications, the authors introduced new ways to calculate the speed as a difference between arrays, or they simply implemented different neighbour search approaches. In the proposed version, the neighbour search is partially readjusted. Each particle has *(i)* a current solution *current(t)*, *(ii)* an intention *pintention(t)* constituted by a random solution, *(iii)* the greedy solution *pgreedy*, *(iv)* the personal best solution *pbest(t)*. Moreover it knows the global best solution found so far *gbest(t)*. To each of the solutions that the particle knows, except for the current solution, a weight is assigned (i.e. w_1 for the *pintention(t)*, w_2 for the *pgreedy*, w_3 for the *pbest(t)*, and w_4 for the *gbest(t)*). During each iteration, each particle explores a new solution, then all particles share with others their *pbest(t)*. If a *pbest(t)* is better than the *gbest(t)*, it is made the new global best. In order to better understand the proposed neighbour search, the authors would like to induce the following notation. Given i a specific picking position and t the current iteration, the authors define:

-$(i, j_1)_t$ the edge which connect i to the next picking position in the *pintention(t)* in iteration t;

-(i, j_2) the edge which connect i to the next picking position in the *pgreedy*;

-$(i, j_3)_t$ the edge which connect i to the next picking position in the *pbest(t)* in iteration t;

-$(i, j_4)_t$ the edge which connect i to the next picking position in the *gbest(t)* in iteration t.

Subsequently, are defined $d_1(t)$ the length of $(i, j_1)_t$, d_2 the length of (i, j_2), $d_3(t)$ the length of $(i, j_3)t$, and $d_4(t)$ the length of $(i, j_4)_t$.

Hence, in each iteration t, each particle explores a new solution by changing its current solution *current(t)*. The new solution *current(t+1)* is built position after position, beginning from $i=1$ and going on until its completion. In particular, the next picking position is selected between j_1, j_2, j_3, and j_4. The selection is made by using a roulette wheel, where the probability to choice the next picking position depends on its distance from i, namely $d_1(t)$, d_2, $d_3(t)$, and $d_4(t)$: bigger is the distance, lower is the probability. More in detail, given t the current iteration and $m \in (1,2,3,4)$, the probability p_m to choice j_m as next node is calculated as in eq.(4).

$$p_m = \begin{cases} \frac{k_m w_m e^{T/d_m}}{\sum_{s=1}^{4} w_s e^{T/d_s}}, & if\ d_m > 0 \\ k_m, & if\ d_m = 0 \end{cases} \quad (4)$$

where $T = \sum_{s=1}^{4} d_s$ and k_m is a binary parameter which defines the possibility to choose j_m as next picking position. Thus, given I the set if picking position not included in the new current solution yet, $k_m = 1$ if $(i, j_m)_t$ exists (i.e. i is not the last picking position of the solution observed), and $j_m \in I$. If after calculating the probabilities it comes that the roulette wheel is not possible because $p1=p2=p3=p4=0$, the next picking position is randomly selected from the set I. After defining the new current solution *current(t+1)*, according to Zhong, Zhang and Chen (2007) a mutation is carried out with low probability. The mutation is made by using the well-known 2-opt algorithm.

4. Setting of parameters

Concerning the ACO, since a design of experiments was already carried out in a recent publication concerning a similar problem (De Santis et al., 2018), the set of parameters defined in that publication is used. Conversely, concerning the PSO, an empirical design was carried out to find out the best combination. The algorithm was tested on 40 randomly generated picking lists made of 20 picking locations. The algorithm was iterated 10 times per each picking list, and the combination of parameters which was providing the best average result on most of the picking lists was selected. The layout considered is the same adopted in the case study. The final optimal set of parameters is the following: number of particles = 40, $w_1 = w_3 = w_4 = 1$, $w_2 = 0.1$, mutation probability = 0.1.

5. Case study

5.1. The assumptions

The comparisons were made doing following assumptions: *(i)* no capacity limit for pickers is considered; *(ii)* every picking list is associated to one and only one order; *(iii)* the improvement strategies are executed singularly on each picking list; *(iv)* possible physical obstructions between pickers are not considered; *(v)* activities to refill the storage locations are not considered; *(vi)* when a picker has visited all the locations associated to a picking list, before taking care of the next one, he/she must go back to the I/O point, thus, given $(i = 1,\ldots,n)$ the set of picking positions to visit, it is always true that $1 = n = I/O$.

5.2. The layout

In the selection of the layout, a real industrial case was observed. The considered warehouse is made of 2 blocks divided by a cross-aisle. In each block there are 7 aisles, with 11 storage locations on each side and 2 aisles at the two opposite ends with 11 storage locations only on one side. Given u the distance unit, each storage location is $2u$ deep and $2u$ large, aisles are $4u$ large, and cross aisle is $4u$ large as well. The Input/Output (I/O) point is in bottom left corner. A representation of the warehouse is provided in

Fig. 1. To translate the warehouse layout into a distance matrix reporting the minimum path between locations, the well-known Floyd-Warshall (FW) algorithm was used. FW algorithm is an exhaustive procedure, which compares each possible path between two given locations (or nodes), in order to find the minimum. Of course, before running FW, additional nodes were appended where the aisles cross each other, otherwise FW would have not worked (Pansart et al. 2018).

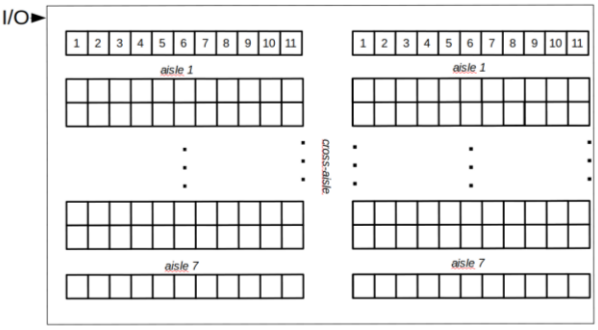

Figure 1: layout considered in the case study above.

5.3. The comparison

The 5 investigated procedures were compared for 3 different complexities of the problem (i.e. picking list of 10 items, picking list of 20 items and picking list of 30 items) and 2 different location assignment policies (i.e. random and class-based). In Tab. 1 and Tab. 2 the comparison of the 5 procedures is carried out respectively in case of random and class-based location assignment policy.

The length of the picking list is of 10 items. For each case, 10 picking lists were used, and the length of the route provided by each algorithm for each list is reported in the tables. Concerning the metaheuristics, both were tested 10 times on each picking list, and in the table the average result and the standard deviation are reported. Observing the results is clear that both metaheuristics always outperformed the other procedures, and both are characterized by great accuracy, since the standard deviation is null. By observing the S-Shape, Largest Gap and Combined policies, some interesting considerations can be made. As it is clearly visible in Fig. 1, in case of random location assignment, the combined method outperforms the others in most tests. Conversely, in case of class-based location assignment, as represented in Fig. 2, the results lead to a different consideration. Since a class-based assignment policy leads by itself to a shorter route, the difference between the analysed routing policies is less evident. However, in a relevant number of tests the largest gap policy outperformed the combined one, and, even the s-shape is providing results closer to those that the combined is able to guarantee. Because of this, and due to relatively easier implementation of the s-shape and largest gap policies, they may be considered more convenient than the combined policy.

Table 1: Results for picking list of 10 items and random location assignment policy

Picking list	ACO		PSO		S-Shape	LG	Combined
	Avg.	Dev.St	Avg.	Dev.St			
1	236	0	236	0	324	304	272
2	160	0	160	0	270	198	208
3	220	0	216	0	324	296	264
4	200	0	200	0	368	346	232
5	196	0	196	0	256	280	222
6	192	0	192	0	358	236	234
7	212	0	212	0	348	274	248
8	196	0	196	0	276	276	230
9	200	0	200	0	324	304	210
10	172	0	172	0	256	240	210

Table 2: Results for picking list of 10 items and class-based location assignment policy

Picking list	ACO		PSO		S-Shape	LG	Combined
	Avg.	Dev.St	Avg.	Dev.St			
1	104	0	104	0	114	112	112
2	152	0	152	0	240	178	196
3	160	0	160	0	180	212	174
4	168	0	168	0	258	222	228
5	144	0	144	0	246	170	285
6	180	0	180	0	248	210	208
7	200	0	200	0	324	230	318
8	164	0	164	0	256	218	180
9	200	0	200	0	256	240	212
10	180	0	180	0	264	220	208

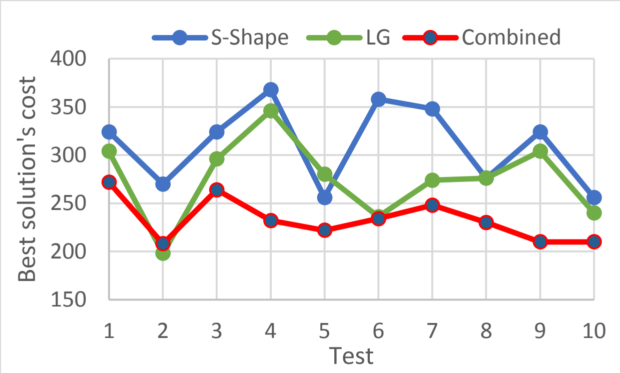

Figure 2: Comparison of heuristics under random assignment.

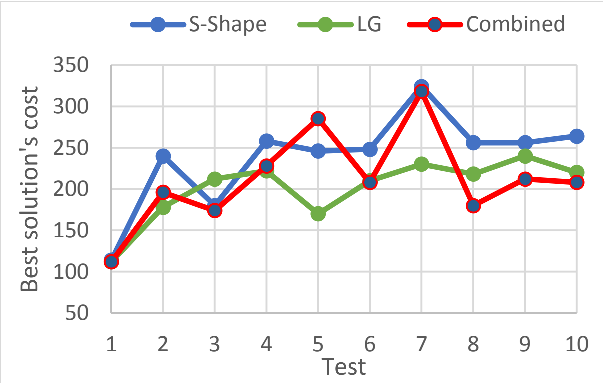

Figure 3: Comparison of heuristics under class-based assignment.

In view of provided results, metaheuristic approaches are still the most efficient and effective. Both outperformed the other routing strategies in 100% of tests carried out, and both had an approximate accuracy of 99% returning the same solution in all the 10 executions made per each picking list. For the purpose of this paper it is not relevant, although, given the results and the accuracy, it could be possible to check if the metaheuristics are finding the best solution. In order to provide a further analysis, the heuristic procedures are therefore abandoned, and the metaheuristics are compared for increasing complexity of the problem. At first, the length of the picking list was increased to 20 elements, and the results are reported in Tab. 3 and Tab. 4. Lastly, the length of the picking list was increased again till 30 items, and the results for random and class-based assignment policies are respectively reported in Tab. 5 and Tab. 6. For reasons of spaces and formatting all the mentioned tables are reported in Appendix A. For each case described by a specific length of the picking lists and a specific location assignment policy, both algorithms were tested on 20 different picking lists. For each picking list, both algorithms were executed 10 times, hence, in tables are reported the average length of the best route found (i.e. Avg.), the standard deviation (i.e. Dev.St.), and the well-known coefficient of variation defined as the standard deviation divided by the average. Finally, the numerical comparison is computed in the last 2 columns of each table. The aim of the author is to compare two main characteristics: the validity of the route provided and the accuracy of the algorithm. Concerning the accuracy, the authors are aware that a bigger accuracy is not a good indicator of the validity of a metaheuristic, since the computational time should be analysed as well. For instance, if an algorithm had a great accuracy and, under equal conditions, it provided every time the same solution; if its computational time is too long, a less accurate but faster algorithm would be preferable. In the same time the first algorithm is executed once, the second one might be iterated more times, exploring different local optimums to choose the best in the end. However, since the compared algorithms are very similar in terms of computational time, only the accuracy is observed. Moreover, both ACO and PSO, are slower if compared to other metaheuristics, thus, a big accuracy can be considered a symptom of quality.

It is also important to point out that, unlike the ACO, the PSO is very feasible for parallel computing, which is a big advantage in year 2020. Although the authors are not going into it in this paper.

Thus, given the objective to compare validity and accuracy of ACO and the proposed PSO, in the last two columns in Tab. 3, 4, 5, and 6, are computed the difference in percentage between the average result and the variance coefficient for each picking list. Given Avg_{ACO} the average result of the ACO and Avg_{PSO} the average result of the PSO, the percentage difference ΔAvg is computed as in eq.(5).

$$\Delta \text{Avg} = \frac{Avg_{PSO} - Avg_{ACO}}{\max(Avg_{ACO}, Avg_{PSO})} \qquad (5)$$

Hence, in the last two columns in Tab. 3, 4, 5, and 6, a negative value means the PSO provided better results than the ACO, otherwise, the ACO worked better than the PSO.

Analysing the results, no relevant influence of the location assignment policies was detected. Of course, in case of class-based assignment policy the travelled distance is usually shorter, but that is because of the allocation logic, and not because of the routing algorithms. The difference between the two algorithms is minimal. On average the ACO is returning a solution 0.47% better than the PSO in case of picking lists made of 20 items, and 1.59% better in case of picking lists made of 30 items. Thus, even if the difference is minimal, it is possible to highlight a deterioration of the PSO, when the length of the picking list increases. The same trend may be detected looking at the accuracy, and its intuitively represented in Fig. 4. As the length of the picking list increases (i.e. the space of feasible solutions increases), by looking at the variation coefficient, it is possible to see how the accuracy of the PSO decrease faster than in case of the ACO.

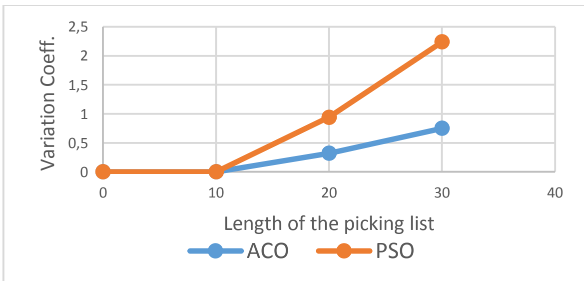

Figure 4: Effect of the length of the picking list on the accuracy of the metaheuristics

To conclude, the authors point out that, in general, a longer picking list increases the number of feasible solutions, but not necessary entails a greater complexity of the problem. To define the complexity of the problem, the number of aisles and the arrival frequency of orders have to be considered, too. However, in the proposed case study, the layout used is always the same, each order is singularly considered, and the picking lists are randomly generated, but always controlling at the same time that the locations required were not to closed and not all the aisles have to be visited. These assumptions avoid into falling into borderline cases, where the length of the picking list would not have any impact on the complexity of the problem.

6.Conclusions

This paper investigates the impact of several routing strategies on minimization of the travel-time for pickers in a manual warehouse. Three classic well-known strategies such as S-Shape, Largest Gap and Combined were analysed and compared with two meta-heuristics owning to swarm intelligence family. The first meta-heuristic was an Ant Colony Optimization (ACO), and, due to existence of several versions, a recent version proposed in 2018, which claims to outperform the previous ones was selected. Finally, a new readjusted version of the Particle Swarm Optimization (PSO) was proposed. Referring to tests carried out, the PSO was very closed to the ACO and outperformed other procedures. The authors are very close to provide a PSO which, outperforms the ACO as well. In order to guarantee a correct comparison, all the proposed routing strategies were validated on by adopting the same layout, the same picking orders, and the same storage assignment policies. To whom it may concern the opinion of the authors, future extensions are possible in three main directions. *(i)* Further constraints might be introduced (e.g. pickers' basket's capacity, orders due date, etc.), *(ii)* different layout might be introduced to observe their effect on proposed strategies, *(iii)* under the same conditions described in this paper, many other algorithms can be compared.

References

Aouay, S., Jamoussi S., and Y. B. Ayed (2013). "Particle swarm optimization based method for Bayesian Network structure learning," 5th International Conference on Modeling, Simulation and Applied Optimization, pp. 1-6.

Bell, J.E., P.R. McMullen (2004). "Ant colony optimization techniques for the vehicle routing problem." Advanced Engineering Informatics, Vol. 18, No. 1, pp. 41-48.

Bertolini, M., Esposito, G., Mezzogori, D., and Neroni, M. (2019). "Optimizing Retrieving Performance of an Automated Warehouse for Unconventional Stock Keeping Units." Procedia Manufacturing, 39, 1681-1690.

De Santis, R., Montanari, R., Vignali, G., and E. Bottani (2018). "An adapted ant colony optimization algorithm for the minimization of the travel distance of pickers in manual warehouses." European Journal of Operational Research, Vol. 267, pp. 120-137.

Dorigo, M., Maniezzo, V., and A. Colorni (1996). "Ant System: optimization by a colony of cooperating agents." IEEE Transactions on Systems. Man, and Cybernetics, Vol. 26, No. 1, pp. 29-41.

Hall, R. W. (1993). "Distance approximations for routing manual pickers in a warehouse", IIE Transactions, Vol. 20, No. 1, pp. 53-62.

Janssen, C.P., Donker, S.F., Brumby, D.P., Kun, A.L. (2019). "History and future of human-automation interaction." International Journal of Human-Computer Studies, Vol. 131, pp. 99-107.

Kennedy, J., and R. Eberhart (1995). "Particle swarm optimization." Proceedings of the IEEE International Conference on Neural Networks, Vol. 4, pp. 1942-1948.

Malakhov, A., Liu, D., Gorshkov, A., and Wilmarth, T. (2018). "Composable Multi-Threading and Multi-Processing for Numeric Libraries." Scipy Conference.

Marchet, G. (1994). "Confronto tra diverse tipologie di percorsi nei sistemi di picking manuali.", Logistica Management, Ed. TeMi.

Pansart, L., Catusse, N., and H. Cambazard. (2018). "Exact algorithms for the order picking problem." Computers and Operations Research, Vol. 100, pp. 117-137.

Penteado, M. M., and Chicarelli, A. R. (2016). "Logistics activities in supply chain business process." The International Journal of Logistics Management, 27(1), 6-30.

Roodbergen, K. J. and R. De Koster (2001). "Routing order pickers in a warehouse with a middle aisle." European Journal of Operational Research, Vol. 133, No. 1, pp. 32-43.

Shi, X.H., Liang, Y.C., Lee, H.P., Lu, C., and Q.X. Wang (2007). "Particle swarm optimization-based algorithms for TSP and generalized TSP. Information Processing Letters, Vol. 103, pp. 169-176.

Shtovba, S. (2005). "Ant algorithms: Theory and applications." Programming and Computer Software, Vol. 31, No. 4, pp. 167-178.

Suresh, A., Harish K.V., and N. Radhika (2015). "Particle Swarm Optimization over Back Propagation Neural Network for Length of Stay Prediction." Procedia Computer Science, Vol. 46, pp. 268-275.

Zhang, G. Q., & Lai, K. K. (2006). "Combining path relinking and genetic algorithms for the multiple-level warehouse layout problem." European Journal of Operational Research, 169(2), 413-425.

Zhong, W., Zhang J., and W. Chen (2007). "A novel discrete particle swarm optimization to solve traveling salesman problem." IEEE Congress on Evolutionary Computation.

Appendix A.

Table 3. Results for picking list of 20 items and random location assignment policy

Picking list	ACO			PSO			Δ Avg.	Δ Coeff. Var.	Picking list	ACO			PSO			Δ Avg.	Δ Coeff. Var.
	Avg.	Dev.St	Coeff. Var	Avg.	Dev.St	Coeff. Var				Avg.	Dev.St	Coeff. Var	Avg.	Dev.St	Coeff. Var		
1	286	4	1,25%	284	0	0,00%	-0,56%	-1,25%	11	380	0	0,00%	382	4	0,94%	0,42%	0,94%
2	295	4	1,48%	295	4	1,48%	0,00%	0,00%	12	300	0	0,00%	300	0	0,00%	0,00%	0,00%
3	248	0	0,00%	248	0	0,00%	0,00%	0,00%	13	308	0	0,00%	311	4	1,41%	1,03%	1,41%
4	282	4	1,27%	285	3	1,18%	1,12%	-0,10%	14	286	4	1,25%	286	4	1,25%	0,00%	0,00%
5	310	5	1,73%	317	9	2,74%	2,02%	1,01%	15	248	0	0,00%	256	6	2,21%	3,13%	2,21%
6	260	0	0,00%	262	4	1,37%	0,61%	1,37%	16	264	0	0,00%	264	0	0,00%	0,00%	0,00%
7	338	2	0,65%	333	2	0,54%	-1,42%	-0,11%	17	260	0	0,00%	263	4	1,66%	1,22%	1,66%
8	296	0	0,00%	298	2	0,74%	0,54%	0,74%	18	260	0	0,00%	260	0	0,00%	0,00%	0,00%
9	212	0	0,00%	212	0	0,00%	0,00%	0,00%	19	306	2	0,72%	304	0	0,00%	-0,78%	-0,72%
10	290	2	0,75%	290	2	0,75%	0,00%	0,00%	20	268	0	0,00%	271	5	1,92%	1,18%	1,92%

Table 4: Results for picking list of 20 items and class-based location assignment policy

Picking list	ACO			PSO			Δ Avg.	Δ Coeff. Var.	Picking list	ACO			PSO			Δ Avg.	Δ Coeff. Var.
	Avg.	Dev.St	Coeff. Var	Avg.	Dev.St	Coeff. Var				Avg.	Dev.St	Coeff. Var	Avg.	Dev.St	Coeff. Var		
21	252	0	0,00%	252	0	0,00%	0,00%	0,00%	31	260	0	0,00%	260	0	0,00%	0,00%	0,00%
22	244	0	0,00%	244	0	0,00%	0,00%	0,00%	32	316	0	0,00%	316	0	0,00%	0,00%	0,00%
23	312	0	0,00%	315	5	1,65%	1,02%	1,65%	33	236	0	0,00%	237	2	0,76%	0,34%	0,76%
24	304	0	0,00%	306	2	0,72%	0,52%	0,72%	34	276	0	0,00%	276	0	0,00%	0,00%	0,00%
25	304	0	0,00%	308	7	2,25%	1,30%	2,25%	35	300	0	0,00%	307	7	2,14%	2,34%	2,14%
26	288	0	0,00%	283	4	1,55%	-1,67%	1,55%	36	320	0	0,00%	320	0	0,00%	0,00%	0,00%
27	287	2	0,62%	287	2	0,62%	0,00%	0,00%	37	288	0	0,00%	289	2	0,62%	0,28%	0,62%
28	303	2	0,59%	308	3	0,92%	1,56%	0,33%	38	305	2	0,59%	305	2	0,59%	0,00%	0,00%
29	279	2	0,64%	284	7	2,63%	1,69%	1,99%	39	289	3	1,16%	290	4	1,23%	0,55%	0,07%
30	248	0	0,00%	250	4	1,43%	0,64%	1,43%	40	308	0	0,00%	313	7	2,10%	1,53%	2,10%

Table 5: Results for picking list of 30 items and random location assignment policy

Picking list	ACO			PSO			Δ Avg.	Δ Coeff. Var.	Picking list	ACO			PSO			Δ Avg.	Δ Coeff. Var.
	Avg.	Dev.St	Coeff. Var	Avg.	Dev.St	Coeff. Var				Avg.	Dev.St	Coeff. Var	Avg.	Dev.St	Coeff. Var		
1	383	6	1,55%	386	2	0,57%	0,83%	-0,98%	11	322	4	1,11%	328	7	2,28%	1,71%	1,17%
2	342	4	1,04%	346	4	1,03%	1,15%	-0,01%	12	393	2	0,46%	393	5	1,33%	0,00%	0,87%
3	321	2	0,56%	329	7	2,00%	2,43%	1,44%	13	308	0	0,00%	308	0	0,00%	0,00%	0,00%
4	377	3	0,89%	380	6	1,49%	0,84%	0,60%	14	398	5	1,35%	407	8	1,89%	2,16%	0,54%
5	315	2	0,57%	315	5	1,65%	0,00%	1,09%	15	310	7	2,16%	313	9	2,92%	0,77%	0,76%
6	316	0	0,00%	329	16	4,74%	3,89%	4,74%	16	344	6	1,64%	345	11	3,11%	0,23%	1,47%
7	358	4	1,00%	362	7	2,01%	0,88%	1,01%	17	356	0	0,00%	369	9	2,47%	3,47%	2,47%
8	349	2	0,51%	354	6	1,71%	1,58%	1,20%	18	332	0	0,00%	341	4	1,29%	2,58%	1,29%
9	348	0	0,00%	357	6	1,66%	2,47%	1,66%	19	360	0	0,00%	368	6	1,72%	2,17%	1,72%
10	349	5	1,50%	354	7	2,05%	1,58%	0,56%	20	390	7	1,71%	390	9	2,36%	-0,20%	0,65%

Table 6: Results for picking list of 30 items and class-based location assignment policy

Picking list	ACO			PSO			Δ Avg.	Δ Coeff. Var.	Picking list	ACO			PSO			Δ Avg.	Δ Coeff. Var.
	Avg.	Dev.St	Coeff. Var	Avg.	Dev.St	Coeff. Var				Avg.	Dev.St	Coeff. Var	Avg.	Dev.St	Coeff. Var		
21	180	0	0,00%	187	7	3,51%	3,85%	3,51%	31	258	5	1,77%	258	6	2,35%	0,00%	0,58%
22	289	2	0,62%	301	7	2,19%	3,99%	1,57%	32	300	0	0,00%	306	10	3,14%	2,09%	3,14%
23	326	2	0,67%	329	11	3,38%	0,97%	2,70%	33	288	0	0,00%	298	2	0,73%	3,49%	0,73%
24	277	2	0,65%	279	2	0,64%	0,86%	-0,01%	34	288	0	0,00%	296	8	2,87%	2,70%	2,87%
25	300	5	1,63%	300	6	1,89%	0,00%	0,25%	35	238	2	0,92%	242	2	0,91%	1,66%	-0,02%
26	317	2	0,56%	315	2	0,57%	-0,51%	0,00%	36	281	2	0,64%	288	10	3,40%	2,50%	2,77%
27	293	2	0,61%	302	12	3,82%	2,92%	3,21%	37	323	7	2,03%	327	17	5,29%	1,22%	3,25%
28	252	0	0,00%	258	5	2,08%	2,48%	2,08%	38	312	3	0,91%	314	13	4,08%	0,76%	3,18%
29	252	6	2,24%	252	9	3,55%	0,00%	1,30%	39	261	2	0,69%	264	4	1,52%	1,21%	0,83%
30	308	0	0,00%	322	13	4,09%	4,23%	4,09%	40	264	0	0,00%	266	4	1,35%	0,60%	1,35%