

University of Parma Research Repository

A hybrid metaheuristic routing algorithm for low-level picker-to-part systems

This is the peer reviewd version of the followng article:

Original

A hybrid metaheuristic routing algorithm for low-level picker-to-part systems / Bottani, E.; Casella, G.; Murino, T.. - In: COMPUTERS & INDUSTRIAL ENGINEERING. - ISSN 0360-8352. - 160:(2021). [10.1016/j.cie.2021.107256]

Availability: This version is available at: 11381/2899812 since: 2021-10-12T11:34:33Z

Publisher: Elsevier

Published DOI:10.1016/j.cie.2021.107256

Terms of use:

Anyone can freely access the full text of works made available as "Open Access". Works made available

Publisher copyright

note finali coverpage

(Article begins on next page)

A hybrid metaheuristic routing algorithm for low-level picker-to-part systems

4 Abstract

An application of an adapted Harmony Search (HS) algorithm is proposed in this study in order to 5 6 minimize manual warehouses' pickers travel distance. Firstly, the distance matrix has been determined 7 through a hybrid algorithm, and then HS is used to compute the pickers' travel distance, developing a 8 MATLAB® simulation model. This model performance is tested on twenty-five scenarios, resulting from 9 variable length of the order pick lists and different manual storage configurations. Thirty picklists are 10 evaluated for each scenario, for a total of 750 simulations. The results provided by the algorithm, 11 compared with those returned by a metaheuristic algorithm and two heuristic routing policies, suggest 12 that HS provides better outputs results than the remaining algorithms. The algorithm is also very efficient 13 from a computational perspective, which allows marking out the pickers route in real-time.

14 Keywords: picking; manual warehouse; routing; travel distance; Floyd-Warshall; Harmony Search.

15 **1 Introduction**

16 Warehouses are typically used for storing or buffering (de Koster, Le-Duc & Roodbergen, 2007) raw 17 materials, WIP, products, consisting of different areas (Roodbergen, Sharp & Vis, 2008; Cao, Jiang, Liu 18 & Jiang, 2018). Supply chain costs are influenced by essential warehouse management activities (Pan, 19 Shih & Wu, 2015). Logistic areas such as shipping, warehousing, receiving, and order picking are crucial 20 to each supply chain (van Gils, Ramaekers, Caris & de Koster, 2018). Among warehouses processes, 21 order picking is the most decisive, as a matter of fact, engraves the total operating costs for 50-70% 22 (Isler, Righetto & Morabito, 2016; Henn & Schmid, 2013; Accorsi, Manzini & Bortolini, 2012; Petersen 23 & Aase, 2004; Hsieh & Tsai, 2006). Typically, a customer's order is converted into a pick list, where 24 the items' location, number, and the picking sequence are detailed. In a manual process, a picker moves 25 into the warehouse, picking and transporting the items from stock, till the central location for packaging 26 and distribution (Hall, 1993; Marchet, Melacini & Perotti, 2015). Among the activities of this process, 27 traveling is the dominant component. Furthermore, travel time has no value for the picking process and 28 is only a cost in terms of a labor hour. Hence, minimizing it is a suitable way for improving the order 29 picking performance (Lu, McFarlane, Giannikas & Zhang, 2016). Routing policies sequence the picklist 30 items to minimize travel times (Roodbergen, Sharp & Vis, 2008). In particular, the pickers' routing 31 through a warehouse is a particular NP-hard traveling salesman problem (TSP) case in which travel is 32 restricted to following aisles (Hall, 1993). In single block storage, different heuristic procedures exist 33 for routing order pickers. In particular, six different strategies – Traversal (also known as S-shape), 34 Return, Midpoint, Largest gap, Combined and Optimal - occur and vary from basic to structured 35 (Petersen, 1997; Dukić & Oluić, 2014). However, although the procedures are very flexibles and simple, 36 optimization algorithms are always the core research (Lu, McFarlane, Giannikas & Zhang, 2016; 37 Petersen, 1999). Optimization problems come up with Heuristic algorithms to find problems better 38 solutions, even if it is not sure to get the optimum (Raouf & Metwally, 2013). Heuristic algorithms are 39 overcome from metaheuristic one, literally intended to find solutions using higher-level techniques 40 (Yang, 2009).

41 For the TSP, few precise algorithms can identify the optimal solution, and, in any case, these algorithms 42 only apply under specific conditions (De Santis, Montanari, Vignali & Bottani, 2018). Nonetheless, 43 (Bouzidi & Riffi, 2014) presented a metaheuristic HS adapted to solve the TSP efficiently. Indeed, the 44 study stated the adaptation efficacy of the HS algorithm related to other methods for solution quality, 45 research time, and results in improvement (i.e., reduction in the percentage of errors). Downstream of 46 these studies, this work proposes an adaptation of the HS metaheuristic algorithm in a manual warehouse 47 to show the adaptability of this Metaheuristic algorithm to pickers' time problem. By comparing the 48 output elaborated by the adapted HS algorithm through the results of the WWO algorithm developed by 49 Bottani, Rinaldi, Montanari, Murino & Centobelli (2016) and with two heuristic algorithms, the paper 50 will also establish that the proposed identify the best pickers path and is computational efficiently.

51 In the remainder of this paper, a deep literature analysis has been conducted about the optimization of 52 the routing manual warehouses, contextualizing the picking process application based on the HS 53 algorithm implementation discussing the most critical aspects in the literature. Then the traditional HS 54 metaheuristic algorithm is described. Hence the designed framework is presented and a numerical 55 example is also proposed to detail the computational procedure in a simple scenario fully. Subsequently 56 the approach is applied to various more complex warehouse configurations to evaluate its capability to 57 get better solutions to the defined problem, and the results returned are discussed. Finally, the study's 58 key findings, discussing the implications, limitations, and suggestions for future research studies are 59 summerized.

60 2 Literature analysis

Routing policies state the order sequence used by the picker to take the requested items off (Grosse &
Glock, 2015). Routing order pickers can easily be interpreted as an alternative to the NP-hard TSP, and
indeed, general TSP model formulations are used for the picking problem (Scholz, Henn, Stuhlmann &
Wäscher, 2016). In simple warehouse layout, fast and exact algorithms for optimal route subsist whilst

for complex storage configurations, no exact algorithm is achievable (Scholz & Wäscher, 2017; Theys 65 66 et al. 2010). The first exact approach was proposed by Ratliff & Rosenthal (1983) using dynamic programming and is valid for a single block warehouse. A 50-aisle problem can be solved in about 1 67 68 minute, and the picking list size does not influence much on the solution time using this procedure. 69 Nowadays, optimal routes can be designated in less than 1 second (Tarczynski, 2013). In order to 70 minimize the pickers travel distance in a warehouse, heuristic algorithms are mostly used, e.g., the so-71 called S-shape (Bahrami, Aghezzaf & Limere, 2017; Roodbergen & de Koster, 2001a). Moving from 72 this consideration, de Koster & Der Poor (1998) have compared the performance of heuristic algorithms 73 and the optimal one. They found that the algorithm of Ratliff & Rosenthal (1983) can be modified in 74 such a way that shortest order picking routes can be found both in centralized and decentralized 75 warehouses. The extended algorithm optimizes in average 25% per travel time route. Roodbergen & de 76 Koster (2001b) have constructed an algorithm, where aisle is variable for the front, the rear, and in the 77 middle, thanks to a cross-aisle.

For difficult layout warehouse configurations, don't exist exact algorithms because the dynamic programming problem is not easy to be generalized for two or more cross-aisles. As a result, heuristic algorithms with added cross-aisle have been found (De Santis et al. 2018, Hall 1993). Theys et al. (2010) have studied the order pickers' route in warehouses with multi parallel aisle. The authors have reformed the TSP applying the Lin-Kernighan-Helsgaun algorithm and reported a 47% lower distance route compared to traditional TSP heuristics.

84 As mentioned above, metaheuristics are intended to find solutions using higher-level modern 85 techniques. Some metaheuristic algorithms have been adapted and applied in the picking problem. To 86 be more precise, Bottani, Cecconi, Vignali & Montanari (2012) have focused on items reallocation to 87 minimize the pickers' path. In particular, the authors formulated a Genetic Algorithm for a new items' 88 allocatio. Batch picking and picker routing problem have been jointly solved by Cheng, Chen, Chen & 89 Yoo (2015) through an innovative hybrid-algorithm consisting of the PSO and the ACO algorithms. The 90 PSO found the best batch picking strategy by minimizing the sum of travel distances, while the ACO 91 searched for the most effective path for each batch. Wisittipanich & Kasemset (2015) elaborated two 92 innovative metaheuristic algorithms - Differential Evolution (DE) and Global Local and Near-Neighbor 93 Particle Swarm Optimization (GLNPSO) – to address warehouse cell optimization in order to minimize 94 the entire travel distances to fill the given picking list. Bottani, Rinaldi, Montanari, Murino & Centobelli 95 (2016) have proposed the more recent WWO algorithm (Zheng, 2015) for identifying the optimal picker routing in a rectangular warehouse. A MATLAB® model was used to optimize the adapted WWO 96 97 algorithm. The authors demonstrated that this study identifies efficiently the shortest pickers' route. 98 Cortés et. al. (2017) have formulated, solving the picking routing problems in medium and large 99 distribution centres. Two TS-hybrid added to a general TS have implemented. The statistical analysis 100 showed that the two-hybrid algorithms presented better results than TS and SA. De Santis et al. (2018)

- introduced an algorithm to optimize the pickers' routing in warehouses. The FW-ACO algorithm
 combined the ACO metaheuristic and the Floyd-Warshall (FW) algorithm. The authors concluded that
- 103 this study added excellent results related to other studies.
- 104 Öztürkoğlu & Hoşer (2017; 2019) have proposed the HS algorithm in the picking field; however, these
- 105 studies did not focus on the routing problem. Instead, the authors have presented a layout design problem
- 106 for composite warehouses. The HS algorithm finds out the tunnel position minimizing the average picker
- 107 travel time in a randomized storage policy case. The authors have used the Harmony Search algorithm
- 108 since more adaptable for design best solutions (Saka et al.2011).
- 109 Because metaheuristic algorithms provide better results than traditional techniques and HS algorithm in
- 110 picking context is poorly discussed, this research focuses on implementing this metaheuristic algorithm
- 111 for the routing problem and to optimize the travel distance and the computational time.

112 **3 The HS algorithm**

The HS algorithm (Geem, Kim & Loganathan 2001) is a metaheuristic population-based method able 113 114 to solve hard and combinatorial or discrete optimization problems (Mansor, Abas, Shibghatullah & 115 Rahman, 2017). HS follows the musical process of a musician who is searching for a perfect harmony (Lee & Geem, 2005). Musical harmony reflects the solution vector, while the musician's improvisations 116 117 reflect the local/global search schemes followed by the algorithm during the optimization. When 118 improvising, a musician can: 1) repeat a famous tune exactly from his/her memory; 2) play something 119 similar to that tune, again on the basis of its memory; or 3) compose a new set of notes randomly. These 120 three processes can be translated into as many options in a quantitative optimization process, namely: 121 1) the usage of harmony memory (HM); 2) the process of pitch adjusting; and 3) randomization (Yang, 122 2009; Geem, Kim & Loganathan, 2001).

- 123 The steps for the application of the HS algorithm are as follows:
- 124 Step 1. Initialization of the problem and parameters setting: harmony memory size (HMS), 125 harmony memory considering rate (HMCR), pitch adjusting rate (PAR) and number of 126 improvisations (NI);
- 127 Step 2. Initialization of the HM;
- Step 3. Improvisation of a new harmony from HM on the basis of memory considerations, pitch
 adjustments, and random selection;
- 130 Step 4. Inclusion of the newly generated harmony in HM if it performs better than the worst131 harmony;
- 132 Step 5. If termination criteria are not satisfied, return to Step 3.
- 133 The overall scheme of the HS algorithm is shown in Figure 1.
- 134 Insert Figure 1

4

HS algorithm was very appropriate to optimize problems like job shop scheduling (Wanga, Pan &
Tasgetiren, 2011), university programs formulation (Al-Betar, Khader & Zaman, 2012; Shahrakia &
Ebrahimib, 2015) and network design (Liu, Yu & Li, 2012; Baskan, 2014; Geem, Tseng & Williams,
2009).

139 3.1 Problem initialization and parameter setting

140 For a minimization problem, the problem is formulated as follows:

Minimize f(x)141 subject to $x_i \in X_i$, i = 1, 2, ..., N142 (1)143 where: 144 f(x) is the objective function; 145 x is a possible solution which typically consists in N decision variables (x_i) ; X_i denotes the possible range of values for each variable, i.e. 146 $X_i = \{x_i(1), x_i(2), \dots, x_i(k)\}$ for discrete decision variables $(x_i(1) < x_i(2) < \dots < x_i(k))\}$ 147 $x_i(K)$; or 148 149 $_{L}x_{i} \leq X_{i} \leq _{U}x_{i}$ for continuous decision variables. In this case, $_{L}x_{i}$ and $_{U}x_{i}$ are the lower and upper bounds for each decision variable, respectively; 150 *K* is the number of possible values for a discrete variable. 151 152 As far as the remaining HS parameters are concerned, HMS is the number of solution vectors (i.e. the

As far as the remaining HS parameters are concerned, HMS is the humber of solution vectors (i.e. the total number of members in the population) in the HM. HMCR is instead a parameter of the improvisation process, used to determine whether the value of a decision variable is to be selected for the solution stored in the HM or randomly chosen from the available range of possible values. PAR is used to determine whether the decision variables are to be adjusted to a neighbor value; finally, NI corresponds to the number of iterations allowed to reach convergence (Al-Betar, Khader & Zaman, 2012; Das, Mukhopadhyay, Roy, Abraham & Panigrahi, 2011).

159 3.2 HM initialization

For initialization purpose, the HM matrix is to be filled with as many randomly generated solutionvectors as the HMS.

$$162 \quad HM = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_{N-1}^1 & x_N^1 \\ x_1^2 & x_2^2 & \dots & x_{N-1}^2 & x_N^2 \\ \dots & \dots & \dots & \dots \\ x_1^{HMS-1} & x_2^{HMS-1} & \dots & x_{N-1}^{HMS-1} & x_N^{HMS-1} \\ x_1^{HMS} & x_2^{HMS} & \dots & x_{N-1}^{HMS} & x_N^{HMS} \end{bmatrix}$$

$$(2)$$

163 The number of rows, in particular, equals the HMS, while the number of columns equals the number of164 variables of each possible solution.

165 3.3 Harmony improvisation from HM

During improvisation, a new harmony vector, $x' = (x'_1, x'_2, ..., x'_N)$, is to be generated from HM based on memory considerations, pitch adjustments, and random selection. In the memory consideration, the value of the first decision variable (x'_1) for the new vector can be chosen from any of the values in the specified HM range $(x'_1 \sim x'_1^{HMS})$. Values of the remaining decision variables (x'_i) can be chosen in the same manner, or, alternatively, new values can be determined using the HMCR parameter, as follows:

171
$$x'_{i} = \begin{cases} x'_{i} \in \{x'_{i}^{1}, x''_{i}^{2}, \dots, x''_{i}^{HMS}\} & with \ probability \ HMCR \\ x'_{i} \in X_{i} & with \ probability \ (1 - HMCR) \end{cases}$$
(3)

Every element of the new harmony vector, $x' = (x'_1, x'_2, ..., x'_N)$, is therefore evaluated to check whether it should be pitch-adjusted. This procedure makes use of the PAR, that sets the rate of adjustment for the pitch chosen from the HM as follows:

175 pitch adjusting decision for
$$x'_i = \begin{cases} Yes & with probability & PAR \\ No & with probability & (1 - PAR) \end{cases}$$
 (4)

176 The value of (1 - PAR) sets the rate of doing nothing. If the pitch adjustment decision for x_i is Yes and

177 x'_i is assumed to be $x_i(k)$, i.e., the k^{th} element in X_i , the pitch-adjusted value of $x_i(k)$ will be:

178 $x'_i = x_i(k+m)$ for discrete variables

179
$$x'_i = x'_i + \alpha$$
 for continuous variables (5)

180 where:

181
$$m \in \{\dots, -2, -1, 1, 2, \dots\}$$
 is the neighboring index;

182 α is the product bw * u;

183 *bw* is an arbitrary distance bandwidth for the continuous design variable; and

184 $u \in [-1; 1]$ is a uniform probability distribution.

185 HMCR and PAR help the algorithm find globally and locally improved solutions, respectively (Afkousi-

186 Paqaleh, Rashidinejad & Pourakbari-Kasmaei, 2010).

187 3.4 HM updating

188 Whenever the new harmony vector $x' = (x'_1, x'_2, ..., x'_N)$ fits the objective function better than the worst

189 harmony vector in the HM, the new harmony will replace the existing worst harmony in the HM.

190 3.5 Termination criterion

191 If the termination criterion (maximum *NI*) is satisfied, the computation stops. Otherwise, step 3 and step192 4 are repeated.

193 4 The proposed approach: adaptation of HS algorithm for picking

194 The framework of the approach proposed in this study is shown in Figure 2. In Table 1 the relating 195 notation is presented.

196 Insert Table 1

From figure 2 it can be seen that the adapted approach includes additional steps compared to the traditional HS metaheuristic, such as some preliminary steps and the implementation of the FW algorithm (cf. (De Santis, et al., 2018)). In particular, this latter algorithm is required to implement and develop the HS algorithm for searching the shortest distance in different warehouses configurations. For clarity, a description of the main steps of the approach is provided in the section that follows; for further

202 details about graphical storage depiction the reader is referred to (De Santis, et al., 2018).

203 Insert Figure 2

204 4.1 Warehouse layout structure

205 The layout structure consists of several picking aisles that have storage locations on both sides. Order 206 pickers can change pick aisle using the cross-aisles positioned perpendicular to the aisles themselves 207 (Roodbergen, Sharp & Vis, 2008). Every time cross-aisles are present, the number of cross aisles equals 208 the number of blocks plus one (Roodbergen & de Koster, 2001a). The main advantage of having extra 209 cross-aisles in a warehouse is the increased number of routing options, resulting in lower travel distance 210 (Vaughan & Petersen, 1999). Three blocks with five aisles with 6 storage locations per aisle side is 211 represented in Figure 3. Solid black squares in the figure indicate the exemplary positions in the rack – 212 picking location - from which items have to be picked (Roodbergen & Vis, 2006).

213 Insert Figure 3

214 4.2 Model hypothesis

- 215 The proposed model is explained through the following hypotheses:
- Multi-block rectangular warehouse;
- No vertical movements to pick up items (i.e., low-level picking);
- During the picker tour, the direction can be changed;
- Aisles can be traveled in both directions;
- The picking aisle is narrow enough to pick items from both sides without covering additional distance;

- The picker starts from the bottom left corner of the depot and returns back once the picklist is 223 completed (i.e., one picker per picking list);
- The amount of items picked in each picklist never saturates the capacity of the picker; hence, 225 capacity constraints are not considered in modeling the problem.

226 4.3 The FW and HS algorithm

Calculating the shortest path between all vertices in an edge-weighted directed graph through the FW
algorithm implementation (Hougardy, 2010). The FW algorithm, determining the shortest path using
the graph theory, makes use of the "distance matrix", built as follows:

230 Step 1. Initialization. The solution matrix same as the input graph is initialized. At the start 231 point process (h=0), the distance matrix structure is initialized as follows:

232
$$D^{(0)} = (D_{ij}^{0}) \text{ where } D_{ij}^{0} = \begin{cases} d_{ij}, \text{ if a direct route connects node i and j} \\ 0, \text{ if } i = j \\ \infty, \text{ if no direct routes connect node i and j} \end{cases}$$
(6)

233 Step 2. Matrix update. The solution matrix is updated by considering all vertices as an 234 intermediate vertex. A new node is then added for the computation of the shortest path between 235 nodes *i* and *j*. Therefore, the distance matrix is updated to D_{ij}^h applying the following formula:

236
$$D_{ij}^{h} = \min\{D_{ij}^{h-1}, D_{ih}^{h-1} + D_{hj}^{h-1}\} \text{ if } i \neq j$$
(7)

237 D_{ij}^{h} is the nodes i to j updated distance considering h intermediate nodes $\{1, ..., h\}$.

238 Step 3. Checking the termination condition. If h = NT, the algorithm ends. The D_{ij}^{NT} element 239 of the distance matrix is the length of the shortest path from nodes *i* to *j*.

The FW algorithm is an input in the proposed model. It is the set of the total number of nodes (*NT*) indicating the picking positions where to pick up the item requested by the customer. The algorithm generates a NT*NT distance matrix. The FW algorithm was implemented in MATLAB[®], to automatically determine the distance matrices in the different warehouse configurations analyzed.

Starting from the FW algorithm's distance matrix, the next step is to determine the shortest path for agiven picklist through the HS algorithm.

Step 1. The first step is the same of that of the original algorithm described above. In particular,
the optimization problem is defined as follows:

248 Minimize
$$\sum_{j=1}^{np-1} D_{j,j+1}, \quad \forall j = 1, 2, ..., np$$
 (8)

249 Moreover, as mentioned before, the HS algorithm parameters required to solve the optimization 250 problem are specified. A static method has been chosen for setting the parameters' value.

251 Step 2. This is the same as the second step (HM initialization) of the traditional HS procedure.

252 Step 3. This is almost the same as the third step (Harmony improvisation from HM) of the traditional HS procedure. The new harmony vector, $x^{new} = \{x_1^{new}, \dots, x_j^{new}, x_{np}^{new}\}, j =$ 253 $1, \dots, np$, will be generated using memory considerations, pitch adjustments, and random 254 selection. The choice of the values for the decision variables follows the same rules of the 255 harmony improvisation, and in particular any value can be chosen from the specified HM range 256 $(x_1^{new1} \sim x_1^{newHMS})$ or, alternatively, new values can be determined using the HMCR parameter: 257

258
$$x_j^{new} = \begin{cases} x_{i,j}^{HM} \text{ with probability HMCR } (i = rand[1, HMS] \text{ and } j \text{ fixed}) \\ x_{i,j}^{HM} \text{ with probability } 1 - HMCR } (i = rand[1, HMS] \text{ and } j = rand[1, np] \end{cases}$$
(9)

- Then, the components of the new harmony vector, $x^{new} = (x_1^{new}, x_2^{new}, \dots, x_N^{new})$, should be 259 260 analysed to determine whether they should be pitch-adjusted; the procedure for pitch-adjustment 261 is described in eq.10:
- $x_{j}^{new} = \begin{cases} x_{i,j}^{HM} & \text{with probability} & PAR \\ x_{j}^{new} & \text{with probability} & (1 PAR) \end{cases}$ 262 (10)
- As per the traditional HS approach, in case the new harmony vector, x^{new} , fits the 263 Step 4. 264 objective function better than the worst harmony vector in the HM, the new harmony is kept in 265 the HM, while the worst harmony is removed.
- 266 Step 5. If the termination condition (i.e. maximum *NI*) has been reached, the computation stops. 267 Otherwise, the algorithm is repeated stating from steps 3.

4.4 Numerical example 268

269 For the sake of clarity, the application of the proposed approach is shown in a numerical example in this 270 section. For testing purpose, a simple scenario (small warehouse and short picklist) is taken, to allow 271 the computational procedure to be almost entirely reproduced. The chosen warehouse layout consists of 272 2 blocks, with 3 aisles per block and 3 storage locations per aisle side; $k_x=5$ [m] and $k_y=1$ [m] are set for 273 this warehouse. A picklist composed of *np*=7 elements (nodes: 2, 7, 11, 14, 16, 19, 23) is considered.

274 **Insert Figure 4**

- 275 As Figure 4 shows, the graph of this representative warehouse consists of 27 total nodes (NT). The cells
- 276 highlighted to represent the storage locations of items (7) in the picklist. The distance matrix (27*27)generated by the FW algorithm is shown in Table 2. 277
- 278 **Insert Table 2**

279 Once the distance matrix has been obtained, the minimum path is calculated by implementing the HS 280 algorithm.

281 As mentioned above, HMCR and PAR help the HS algorithm find globally and locally improved solutions (Dell'orco, Baskan & Marinelli, 2013). To ensure good performance of the algorithm, Geem, 282

283 2006; Bouzidi & Riffi, (2014) have recommended that HMCR values range from 0.70 to 0.95, 0.20, 284 PAR values from 0.2 to 0.50, and HMS values from 10 to 50. In line with these considerations, and after 285 performing a preliminary series of hand-tuning experiments on the adapted HS algorithm, the 286 parameters were set as follows: HMS=np; HMCR=0.95; PAR=0.45; *NI*=500.

The modified HS algorithm was implemented under the commercial software MATLAB[®]. The simulation procedure was run on an AMD Athlon, 3GHz with 4GB RAM desktop computer equipped with Windows 7 Professional. Once the last iteration has been completed, the HS algorithm returns the following picking sequence, whose path is shown in Figure 5.

 $291 \qquad 0 - 11 - 19 - 23 - 14 - 16 - 7 - 2 - 0$

292 Insert Figure 5

The specific results of the performance evaluation for the HS algorithm, shown for distance, computational time, and convergence, are highlighted in Figure 6 and Table 3.

295 Insert Figure 6

296 Insert Table 3

The results in Table 3 show that the shortest path, for this configuration, is 42 meters, obtained after 241 iterations (see also Figure 6), i.e., less than 5% of the whole set of solutions (7! = 5040) for the problem under examination. Moreover, the computational time required to run the algorithm amounts to 2.51 seconds.

301 **5** Application and discussion

302 5.1 Warehouse layouts

An exhaustive test of performance of the proposed approach was made on five warehouse configurations, obtained by varying the number of blocks (1-5, step 1); length of the order picklist was varied as well (10-50 items, step 10). Twenty-five scenarios (5 sizes of pick lists x 5 warehouse configurations) were examined overall, and 30 different pick lists were tested for each scenario to ensure significance of the results obtained; the total number of simulations was 750.

The experiments were carried out considering a representative warehouse layout, with longitudinal aisles, where shelves are placed on both sides, and with 32 picking positions for each aisle side. In the multi-block layouts, the picking positions (*ppa* and *ppb*) in the sub-aisles of the two- and four-block warehouses are equally distributed and accounts for 16 and 8, respectively. In the three- and five-block configurations, instead, the picking positions are divided differently. In the first case (three blocks), in the sub-aisles of two blocks farthest from the depot, there are ten picking positions, while in the remaining block, there are 12 picking positions. In the five-block layout, there are 6 picking positions

- 315 in the sub-aisles of the four blocks furthest from the depot, while there are 8 picking positions in the 316 remaining block.
- 317 In general, while the total number of picking positions remains the same (i.e., 640) in each warehouse
- 318 layout, the number of NT changes (and in particular increases) as a function of the number of blocks,
- 319 consistently with the increase in the number of cross-aisles and, therefore, of service nodes.
- A rectangular warehouse, with a base of 55 meters and a depth that ranges from 40 to 52 meters depending on the number of blocks, is assumed. The aisle width is 3 meters.

322 5.2 Experimental results

323 As mentioned before, the validation of the HS algorithm results was made by comparing the travel 324 distance obtained with that resulting from the application of one metaheuristic algorithm (i.e., WWO 325 algorithm) and two traditional routing policies (i.e., S-shape and largest gap). The WWO was chosen as 326 a suitable algorithm for benchmarking the results of the proposed approach as WWO proved to be 327 always able to identify the global optimal solution in the tests carried out by Bottani, Rinaldi, Montanari, Murino & Centobelli (2016). Table 4 reports the results of the proposed approach in terms of distance 328 329 travelled and computational time, depending on the warehouse configuration and problem complexity; 330 these outcomes were obtained with the parameters settings detailed in Section 4. In Table 4, the 331 percentage of the standard deviation of the outcomes is also reported. Data in bold highlight the best 332 result(s) obtained for each scenario, as well as the algorithm(s) that returned the most effective 333 solution(s).

334 Insert Table 4

335 5.3 Discussion

From the results in Table 4, the following primary considerations emerge. In terms of the picking 336 distance, it is evident that the HS and WWO algorithms provide almost identical results. In particular, 337 338 the HS algorithm generates better solutions in 18 configurations out of 25, compared to 7 for the WWO 339 algorithm. To be more precise, as can be seen from Table 4, with ten order lines the HS algorithm 340 provided slightly worse results than the WWO (i.e., 200.60 vs. 200.00 meters) in one configuration only, 341 i.e., the three-block warehouses; the same consideration holds true for order lines of 30 and 50 items. 342 With order lines of 20 or 40 items, instead, the WWO algorithm turned out to be better than the HS in 343 two configurations (i.e., the three- and four-block warehouses). Nonetheless, the travel distance returned 344 by the HS is better than that of the WWO algorithm by approximately 0.37% on average. In four- and 345 five-block configurations, the improvement is more significant, reaching 0.55% and 0.59%, 346 respectively. Moreover, in five-block warehouses, the HS approach generates solutions that are always 347 better than those of the WWO algorithms.

These outcomes do not contradict the results reported in Bottani, Rinaldi, Montanari, Murino & Centobelli (2016). Indeed, although these authors found that WWO was always able to find the optimal solution in their testing scenarios, the configurations tested referred to one-block warehouses only, while no tests were proposed for multiple-blocks warehouses. Therefore, the outcomes of the present study rather complement the findings previously available and allow us to argue that the HS approach overcomes the WWO algorithm for complex warehouse configurations.

354 Outcomes also show that the performance of the two metaheuristics varies as a function of the picklist 355 size. In general terms, the HS overcomes the WWO algorithm, with a peak of 1.05% reduction in the 356 length of the picking tour for pick lists of 30 items. A greater size of the picking list involves a lower 357 difference in the performance of the two algorithms (0.23% and 0.05% respectively for 40 and 50 items in the picking list). The standard deviation of the calculated distances decreases as well: this is probably 358 359 due to the fact that with more items in the picklist, the positions of items become closer in the warehouse, 360 so that the tour is almost defined and the heuristic algorithms have less room for shortening the total 361 travel distance. On the contrary, for small pick lists, items to be picked are sparse in the warehouse, so 362 that their specific picking position and the way it is reached can make the difference in terms of the total 363 distance travelled.

Compared to the remaining heuristic routing policies, it is immediate to see that the travel distance returned by the modified HS algorithm is always shorter; this result was expected (and obviously desirable); in fact, to prove its effectiveness, it is almost essential that a newly proposed metaheuristic algorithm overcomes at least the performance of the heuristic routing policies. The results obtained show that the modified HS approach generates a travel distance, which, on average, is 26.90% and 11.46% shorter than that obtained by applying the S-shape and largest gap policies, respectively.

With respect to the computational time, results show once again that the performance of the HS algorithm is much better than that of the WWO algorithm. In particular, HS shows an average computational time approximately 24% lower than that of WWO. This effective performance can be attributed to the quite simple structure of the HS algorithm as well as to its combination with the FW approach, which in previous studies (e.g. De Santis, Montanari, Vignali & Bottani, 2018) was demonstrated to enhance the performance of metaheuristic algorithms.

376 6 Conclusions

This study has proposed an adapted approach to reduce the picking distance in manual warehouses. To be more precise, this paper has: 1) suggested the combination of the HS metaheuristic algorithm with the FW one; 2) shown its application to the picking problem in a manual warehouse; and 3) tested its performance in terms of travel distance and computational time. The adapted approach includes some preliminary steps, which basically refer to the implementation of the FW algorithm; this latter was applied as a useful approach to mathematically reproduce the different warehouse configurations and to preliminarily derive the shortest distance between each pair of nodes in the warehouse. Then, the proposed framework includes 5 steps that reflect the logic of the traditional HS algorithm; this latter is used to determine the shortest distance for each picking tour in the various warehouse configurations. All steps were coded in MATLAB[®] to be run automatically.

The implementation of the proposed approach was first shown with respect to a typical warehouse layout, simple enough to allow the detailed description of all the steps of the procedure. The algorithm performance was then tested on five different warehouse configurations, with variable number of blocks and picking list size. Twenty-five scenarios were considered overall, with 30 random picking lists for each of them, for a total of 750 simulations.

392 From a theoretical perspective, the outcomes obtained highlights how the proposed approach 393 outperforms both the heuristic routing policies and the WWO algorithm in determining the shortest route 394 of pickers. Moreover, by analysing the computational time, it is easy to deduce that the HS algorithm 395 adds quality compared to some well-known heuristic policies and to the WWO algorithm. In summary 396 this study has proposed a metaheuristic hybrid algorithm whose results encourage its application in 397 practice. Besides, the approach proposed in this paper contains a set of additional steps compared to the 398 traditional HS algorithm, which enhance its effectiveness in minimising the travel distances of pickers 399 in warehouses. From a practical perspective, this paper focuses on manual warehouses and has been 400 tested in some selected configurations. Nonetheless, this study can be implemented in additional layouts 401 or configurations, to test its performance in further scenarios. As the proposed approach was effective 402 in improving the order picking performance in the scenarios tested, it is expected to provide interesting 403 outcomes in different configurations too.

404 Although the outcomes of this paper can be seen as of general validity, this paper has some limitations 405 that should be mentioned. As an example, in this study, random storage of items in the warehouse was 406 assumed; however, for picking lists of small sizes it would probably be preferable to use a class-based 407 storage policy, to further decrease the travel distance. For picking lists of greater size, instead, a random 408 storage policy is likely to provide results similar to the class-based one, which suggests that testing this 409 latter policy would not be essential. Moreover, in this study, the picker starts from the receiving area 410 and returns to the same place once he has picked the full set of items in the picking list; however, for 411 order pick lists of 40 or 50 items, it would be appropriate to include the capacity of the picker as a 412 constraint of the problem. To this end, it could be interesting to apply a multi-objective optimization 413 procedure to reduce the travel distance and maximize the saturation of the picker's capacity, to evaluate 414 whether (and to what extent) the capacity of the picker could affect the travel distance. Moreover, further 415 research might take into consideration a different type of layouts, with a particular attention to non-416 conventional warehouses (Fishbone, U-Shaped, and Flying-V). Indeed, changing the warehouse layout

- would certainly involve variations in the distance travelled and in the time taken to complete a pickingtour, which could lead to additional insights. Further research may also concern on the presence of
- 419 different width of aisles (wide aisle o ultra-narrow aisle), which would lead to congestion in the aisles.

420 **References**

- 421
- R. de Koster, T. Le-Duc e K. Roodbergen, «Design and control of warehouse order picking: a literature review,» *European Journal of Operational Research*, vol. 182, n. 2, pp. 481-501, 2007.
- [2] K. Roodbergen, G. Sharp e I. Vis, «Designing the layout structure of manual order picking areas in warehouses,» *IIE Transactions (Institute of Industrial Engineers)*, vol. 40, n. 11, pp. 1032-1045, 2008.
- [3] W. Cao, P. Jiang, B. Liu e K. Jiang, «Real-time order scheduling and execution monitoring in public warehouses based on radio frequency identification,» *International Journal of Advanced Manufacturing Technology*, vol. 95, n. 5-8, pp. 2473-2494, 2018.
- [4] J.-H. Pan, P.-H. Shih e M.-H. Wu, «Order batching in a pick-and-pass warehousing system with group genetic algorithm,» *Omega*, vol. 57, pp. 238-248, 2015.
- [5] T. van Gils, K. Ramaekers, A. Caris e R. de Koster, "Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review," *European Journal* of Operational Research, vol. 267, pp. 1-15, 2018.
- [6] C. Isler, G. Righetto e R. Morabito, «Optimizing the order picking of a scholar and office supplies warehouse,» *International Journal of Advanced Manufacturing Technology*, vol. 87, n. 5-8, pp. 2327-2336, 2016.
- [7] S. Henn e V. Schmid, «Metaheuristics for order batching and sequencing in manual order picking systems,» *Computers & Industrial Engineering*, vol. 66, pp. 338-351, 2013.
- [8] R. Accorsi, R. Manzini e M. Bortolini, «A hierarchical procedure for storage allocation and assignment within an order-picking system. A case study,» *International Journal of Logistics Research and Applications*, vol. 15, n. 6, pp. 351-364, 2012.
- [9] C. Petersen e G. Aase, «A comparison of picking, storage, and routing policies in manual order picking,» *International Journal of Production Economics*, vol. 92, n. 1, pp. 11-19, 2004.

- [10] L.-F. Hsieh e L. Tsai, «The optimum design of a warehouse system on order picking efficiency,» *The International Journal of Advanced Manufacturing Technology*, vol. 28, n. 5-6, pp. 626-637, 2006.
- [11] R. Hall, «Distance approximations for routing manual pickers in a warehouse,» *IIE Transactions* (*Institute of Industrial Engineers*), vol. 25, n. 4, pp. 76-87, 1993.
- [12] G. Marchet, M. Melacini e S. Perotti, «Investigating order picking system adoption: a case-study-based approach,» *International Journal of Logistics Research and Applications*, vol. 18, n. 1, pp. 82-98, 2015.
- [13] W. Lu, D. McFarlane, V. Giannikas e Q. Zhang, «An algorithm for dynamic order-picking in warehouse operations,» *European Journal of Operational Research*, vol. 248, pp. 107-122, 2016.
- [14] C. Petersen, «An evaluation of order picking routeing policies,» International Journal of Operations & Production Management, vol. 117, n. 11, pp. 1098-1111, 1997.
- [15] G. Dukić e Č. Oluić, «Order-picking routing policies: simple heuristics, advanced heuristics or optimal algorithm,» *Journal of Mechanical Engineering*, vol. 50, n. 11, pp. 530-535, 2014.
- [16] C. Petersen, «The impact of routing and storage policies on warehouse efficiency,» International Journal of Operations & Production Management, vol. 19, n. 10, pp. 1053-1064, 1999.
- [17] O. Raouf e M.-B. Metwally, «A survey of harmony search algorithm,» *International Journal of Computer Applications*, vol. 70, n. 28, pp. 17-26, 2013.
- [18] X.-S. Yang, «Harmony search as a metaheuristic algorithm,» Studies in Computational Intelligence, vol. 191, pp. 1-14, 2009.
- [19] R. De Santis, R. Montanari, G. Vignali e E. Bottani, «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, 2018.
- [20] M. Bouzidi e M. Riffi, «Adaptation of the harmony search algorithm to solve the travelling salesman problem,» *Journal of Theoretical and Applied Information Technology*, vol. 62, n. 1, pp. 154-160, 2014.
- [21] E. Bottani, M. Rinaldi, R. Montanari, T. Murino e P. Centobelli, «An adapted water wave optimization algorithm for routing order pickers in manual warehouses,» in *Proceedings of the Summer School Francesco Turco*, 2016.

- [22] H. Grosse e C. Glock, «The effect of worker learning on manual order picking processes,» International Journal of Production Economics, vol. 170, pp. 882-890, 2015.
- [23] A. Scholz, S. Henn, M. Stuhlmann e G. Wäscher, «A new mathematical programming formulation for the single-picker routing problem,» *European Journal of Operational Research*, vol. 253, pp. 68-84, 2016.
- [24] A. Scholz e G. Wäscher, «Order batching and picker routing in manual order picking systems: the benefits of integrated routing,» *Central European Journal of Operations Research*, vol. 25, n. 2, pp. 491-520, 2017.
- [25] C. Theys, O. Bräysy, W. Dullaert e B. Raa, «Using a TSP heuristic for routing order pickers in warehouses,» *European Journal of Operational Research*, vol. 200, n. 3, pp. 755-763, 2010.
- [26] H. Ratliff e A. Rosenthal, «Order-Picking in a rectangular warehouse: a solvable case of the traveling,» *Operations Research*, vol. 31, n. 3, pp. 507-521, 1983.
- [27] G. Tarczynski, «Warehouse real-time simulator how to optimize order picking time,» *SSRN Electronic Journal*, 2013.
- [28] B. Bahrami, E.-H. Aghezzaf e V. Limere, «Using simulation to analyze picker blocking in manual order picking systems,» *Procedia Manufacturing*, vol. 11, pp. 1798-1808, 2017.
- [29] K. Roodbergen e R. De Koster, «Routing methods for warehouses with multiple cross aisles,» International Journal of Production Research, vol. 39, n. 9, pp. 1865-1883, 2001.
- [30] R. de Koster e E. Der Poor, «Routing orderpickers in a warehouse: a comparison between optimal and heuristic solutions,» *IIE Transactions*, vol. 30, pp. 469-480, 1998.
- [31] E. Bottani, M. Cecconi, G. Vignali e R. Montanari, «Optimisation of storage allocation in order picking operations through a genetic algorithm,» *International Journal of Logistics: Research and Applications*, vol. 15, n. 2, p. 127–146, 2012.
- [32] C.-Y. Cheng, Y.-Y. Chen, T.-L. Chen e J.-W. Yoo, «Using a hybrid approach based on the particle swarm optimization and ant colony optimization to solve a joint order batching and picker routing problem,» *Internation Journal of Production Economics*, vol. 170, pp. 805-814, 2015.
- [33] W. Wisittipanich e C. Kasemset, «Metaheuristics for warehouse storage location assignment,» Chiang Mai University Journal of Natural Sciences, vol. 14, n. 4, pp. 361-377, 2015.

- [34] Y.-J. Zheng, «Water wave optimization: A new nature-inspired metaheuristic,» *Computers and Operations Research*, vol. 55, pp. 1-11, 2015.
- [35] P. Cortés, R. Gómez-Montoya, J. Muñuzuri e A. Correa-Espinal, «A tabu search approach to solving the picking routing problem forlarge- and medium-size distribution centres considering theavailability of inventory and K heterogeneous material handlingequipment,» *Applied Soft Computing Journal*, vol. 53, pp. 61-73, 2017.
- [36] Ö. Öztürkoğlu e D. Hoşer, «A polynomial time tour algorithm for order picking operations in warehouses and new aisle designs,» in *Proceedings of the International Conference on Industrial Engineering and Operations Management*, Rabat, Marocco, 2017.
- [37] Ö. Öztürkoğlu e D. Hoser, «A discrete cross aisle design model for order-picking warehouses,» European Journal of Operational Research, vol. 275, n. 2, pp. 411-430, 2019.
- [38] M. Saka, I. Aydogdu, O. Hasancebi e Z. Geem, «Harmony search algorithms in structural engineering,» *Studies in Computational Intelligence*, vol. 359, pp. 145-182, 2011.
- [39] Z. Geem, J. Kim e G. Loganathan, «A new heuristic optimization algorithm: harmony search,» *Simulation*, vol. 76, pp. 60-68, 2001.
- [40] N. Mansor, Z. Abas, A. Shibghatullah e A. Rahman, «Modified parameters of harmony search algorithm for better searching,» in *IOP Conference Series: Materials Science and Engineering*, Melaka (Malaysia), 2017.
- [41] K. Lee e Z. Geem, «A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice,» *Computer Methods in Applied Mechanics and Engineering*, vol. 194, n. 36-38, pp. 3902-3933, 2005.
- [42] L. Wanga, Q.-K. Pan e M. Tasgetiren, «A hybrid harmony search algorithm for the blocking permutation flow shop scheduling problem,» *Computers & Industrial Engineering*, vol. 61, n. 1, pp. 76-83, 2011.
- [43] M. Al-Betar, A. Khader e M. Zaman, «University course timetabling using a hybrid harmony search metaheuristic algorithm,» *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, n. 5, pp. 664-681, 2012.
- [44] A. Shahrakia e S. Ebrahimib, «A new approach for forecasting enrollments using harmony search algorithm,» *Journal of Intelligent & Fuzzy Systems*, vol. 28, n. 1, pp. 279-290, 2015.

- [45] L. Liu, H. Yu e L. Li, «Distribution network reconfiguration based on harmony search/genetic hybrid algorithm,» in *China International Conference on Electricity Distribution, CICED*, Shanghai (China), 2012.
- [46] O. Baskan, «Harmony search algorithm for continuous network design problem with link capacity expansions,» *KSCE Journal of Civil Engineering*, vol. 18, n. 1, pp. 273-283, 2014.
- [47] Z. Geem, C.-L. Tseng e J. Williams, «Harmony search algorithms for water and environmental systems,» *Studies in Computational Intelligence*, vol. 191, pp. 113-127, 2009.
- [48] S. Das, A. Mukhopadhyay, A. Roy, A. Abraham e B. Panigrahi, «Exploratory power of the harmony search algorithm: Analysis and improvements for global numerical optimization,» *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* vol. 41, n. 1, pp. 89-106, 2011.
- [49] M. Afkousi-Paqaleh, M. Rashidinejad e M. Pourakbari-Kasmaei, «An implementation of harmony search algorithm to unit commitment problem,» *Electrical Engineering*, vol. 92, n. 6, pp. 215-225, 2010.
- [50] K. Roodbergen e R. de Koster, «Routing order pickers in a warehouse with a middle aisle,» European Journal of Operational Research, vol. 133, n. 1, pp. 32-43, 2001.
- [51] T. Vaughan e C. Petersen, «The effect of warehouse cross aisles on order picking efficiency,» International Journal of Production Research, vol. 37, n. 4, pp. 881-897, 1999.
- [52] K. Roodbergen e I. Vis, «A model for warehouse layout,» IIE Transactions (Institute of Industrial Engineers), vol. 38, n. 10, pp. 799-811, 2006.
- [53] S. Hougardy, «The Floyd–Warshall algorithm on graphs with negative cycles,» *Information Processing Letters*, vol. 110, p. 279–281, 2010.
- [54] M. Dell'orco, O. Baskan e M. Marinelli, «A harmony search algorithm approach for optimizing traffic signal timings,» *Promet - Traffic - Traffico*, vol. 25, n. 4, pp. 349-358, 2013.
- [55] Z. Geem, «Optimal cost design of water distribution networks using harmony search,» *Engineering Optimization*, vol. 38, n. 3, pp. 259-277, 2006.

422

423