An adapted water wave optimization algorithm for routing order pickers in manual warehouses

Eleonora Bottani*, Marta Rinaldi*, Roberto Montanari*, Teresa Murino**, Piera Centobelli**

* Department of Industrial Engineering, University of Parma, viale G.P.Usberti 181/A, 43124 Parma – Italy (Eleonora.bottani@unipr.it; Marta.rinaldi@unipr.it; Roberto.montanari@unipr.it)

** Department of Chemical, Materials and Industrial Production Engineering, University of Naples “Federico II”, Piazzale Tecchio 80, 80125 - Napoli – Italy (teresa.murino@unina.it, piera.centobelli@unina.it)

Abstract: In this paper, we present a new algorithm for routing order pickers in manual warehouses and optimizing the total distance covered. The proposed approach is an adaptation of the recent water wave optimization (WWO) algorithm, a nature-inspired algorithm based on the shallow water wave theory. The algorithm is tested on a rectangular 1-block warehouse, with 12 aisles and 45 storage locations per aisle side (1080 storage locations overall). The depot is located at the bottom left corner of the warehouse. A Microsoft Excel™ file was programmed to reproduce both the adapted WWO algorithm and to test its performance, in terms of the travel distance covered by pickers. A formal design of experiment (DOE) analysis is carried out on some of the algorithm parameters, to assess their effect on the performance observed (namely computational time and number of iterations required to reach convergence) and identify the optimal setting.

Keywords: picking; manual warehouse; routing; travel distance; water wave optimization.

1. Introduction

Warehouse operation and management is an essential part of manufacturing and service operations (Zhang & Lai, 2006). The efficiency and effectiveness of logistics activities in general and of distribution networks in particular, is largely determined by the operation of the warehouses, as the nodes of these networks. Warehouses are vital links in the supply chain; here, products are temporarily stored and subsequently retrieved from storage locations to fulfil the customer’s orders. The logistics cost relating to warehouse processes, including receiving, storage, order picking and shipping, is often high (Rouwenhorst et al., 2000). Among the different warehouse processes, order picking is generally recognized as the most expensive activity, because it tends to be either very labour intensive or capital intensive (Frazelle, 2002). Order picking is the process of selecting a set of items, retrieving them from their storage locations and transporting them to a sorting/consolidation process for order fulfillment and shipment, in response to a customer’s request (Rouwenhorst, et al., 2000). Typically, orders from customers consist of different order lines, each line representing a unique item, or a stock keeping unit, which is requested in a defined quantity.

The picking process can either be performed manually or (partly) automated. In the case of a manual process, it is estimated that picking operations account for more than 55% of the total cost of warehouse operations (Coyle et al., 1996; Bottani et al., 2015). This high cost is mainly due to the fact that approximately 50% of the total order picking time is spent by pickers in (unproductive) travelling. This part of the order picking time is commonly known as the “travel time” of pickers and affects the total order picking time to the largest extent (Tompkins, et al., 1996). Obviously, the travel time is an increasing function of the travel distance; hence, minimizing the travel distance is suggested by many authors as the main leverage for optimizing the total picking time of warehouses (Jarvis & McDowell, 1991; Hall, 1993; Petersen, 1999; Roodbergen & de Koster, 2001; Petersen & Aase, 2004).

Reducing the travel distance of pickers has a direct impact on warehouse performance in terms of cost and delivery lead-time, and consequently affects the performance of the whole supply chain. Indeed, the faster items are picked from the warehouse, the shorter the time spent on order fulfillment; hence, the lead-time required for delivering the product to the final customer decreases correspondingly (de Koster et al., 2007; Bottani et al., 2012). This is why both researchers and logistics managers consider order picking as one of the most promising areas for productivity improvement (de Koster et al., 2007).

Researchers agree that several factors can affect the travel distance of pickers, including: the overall structure of the warehouse, in terms of size and layout (Parikh & Meller, 2010); the operational strategy, e.g. order picking vs. batch picking (Van Nieuwenhuyse & de Koster, 2009); the storage assignment policy (Petersen & Schmenner; Bottani et al., 2012); the use of zone picking (de Koster et al., 2012); the picker routing (Petersen & Aase, 2004).

In this paper, we focus on the optimization of the routing policy of the picker; this is suggested as one of the most flexible factors, with greater potential for dynamic adjustment (Chen et al., 2013). Moreover, significant savings in the travel time of pickers can be generated when using a dedicated routing heuristic, because in real cases,
the majority of warehouses use very simple routing policies and the picking process is carried out manually (Petersen, 1999).

The identification of the minimum travel distance of the pickers in conventional multi-parallel-aisle warehouses, starting from the set of items to be picked, has been demonstrated to be a combinatorial NP-hard travelling salesman problem (TSP) (Theys et al., 2010). For the solution of this problem, there are very few exact algorithms that can be applied only under specific conditions. Nature-inspired algorithms, however, can offer an interesting alternative to solve complex optimization problems which typically have non-convex and non-linear solution spaces and which are otherwise computationally difficult to solve by conventional mathematical programming methods (de Jong, 2006).

This paper proposes the use of a nature-inspired algorithm, called water wave optimization (WWO), to identify the optimal routing of a picker in a low-level, picker-to-parts warehouse. WWO has been proposed very recently and there are no applications to the picking problem. Nonetheless, Zheng (2015) has tested WWO on a diverse set of benchmark problems and applied it to a real-world high-speed train scheduling problem. The author found that WWO is very competitive with state-of-the-art evolutionary algorithms and concluded that, because of its effectiveness, the new metaheuristic is expected to have wide applications in real-world engineering optimization problems. Moving from these considerations, this paper first shows how the WWO algorithm can be effectively adapted to be used for the optimization of routing in picking problems. By comparing the outcomes generated by the adapted WWO procedure with the optimal solution of the problem, the paper will also demonstrate that proposed approach is effective in identifying the shortest path of pickers and is efficient from a computational point of view.

2. The algorithm

2.1 Background: the WWO procedure

The WWO algorithm (Zheng, 2015) takes inspiration from the shallow water wave theory. The algorithm is based on three phenomena of water waves, namely propagation, refraction and breaking. Each wave is characterised by a wavelength $\lambda \in \mathbb{R}^+$ and a height (amplitude) $h \in \mathbb{Z}^+$, $h \leq h_{\text{max}}$. Each solution $x$, i.e. a wave, is evaluated with respect to its fitness function $f(x)$. In the general case, a wave can have $n$ dimensions. The algorithm consists in the steps described below.

Step 1. The starting point of the algorithm is a set of initial (random) solutions to the problem examined, i.e. a population of waves. $N$ denotes the number of waves in the population. The height of each wave is initialized at $h_{\text{max}}$.

Step 2. The fitness function $f(x)$ of each wave $x$ is computed and the best one $x^*$ is identified.

Step 3. Each wave is initially subject to propagation, once per each iteration of the algorithm. Propagating the wave means shifting it from its original position to a new one, according to the following formula:

$$x'(d) = x(d) + \text{rand}(-1; 1) \cdot \lambda L(d)$$

(1)

where $\text{rand}(-1; 1)$ denotes a random number with uniform distribution between -1 and 1, $d (1 \leq d \leq n)$ is a generic dimension of the problem and $L(d)$ is the length of the $d$ dimension of the search space. After propagation, the fitness of the new wave $f(x')$ should be calculated and compared to the original one. If $f(x') > f(x)$ (for a maximization problem) the new wave $x'$ will replace $x$ in the population and the height of the new wave will be reset at $h_{\text{max}}$.

Step 4. The next step is refraction. Refraction is carried out only on those waves whose height has decreased to 0. For these waves, which are destined to disappear, a new position is randomly identified (this basically corresponds to the generation of a new wave). The following formula is used:

$$x'(d) = N\left(\frac{x(d) + x(d)}{2}, \frac{|x'(d) - x(d)|}{2}\right)$$

(2)

where $N$ denotes a normal distribution (with parameters $\mu$ and $\sigma$) and $x^*$ is the best solution currently available in the population. After refraction, the height of the wave is reset at $h_{\text{max}}$, while its wavelength is set at $\lambda' = \frac{L(x)}{f(x')}$.

Step 5. The final step, namely breaking, is carried out only on waves that have found the best solutions to the problem examined, i.e. on waves $x$ that became $x'$ after propagation. Breaking is a local search around $X$ and involves a random group of $k (1 \leq k \leq k_{\text{max}} \leq n)$ dimensions of the problem. For the generic $d$ dimension, the following computation is carried out:

$$x'(d) = x(d) + N(0; 1) \beta L(d)$$

(3)

where $\beta$ is the breaking coefficient. If the breaking generated a wave $x'$ better than $x'$, $x'$ will replace $x^*$ in the population. Otherwise, $x'$ is retained.

At each iteration of the algorithm, steps 2-5 are repeated on the whole set of waves in the population. The algorithm typically stops when a given number of iterations $I$ has been carried out.

2.2 The adapted WWO algorithm for picking

The WWO algorithm cannot be applied to the picking
problem in its original form. Rather, some modifications should be made to adapt it to the problem investigated.

We recall that we are studying the routing of pickers in manual warehouses. In this kind of problem, optimization algorithms are typically used to generate a sequence of picking locations that allows the total travel distance to be minimised, i.e. to identify the optimal sequencing of items in the picking list. Obviously, the picking list contains always the same items, which can only be changed in the order they are listed. Also, any element should appear exactly once. Accordingly, the possible solutions (i.e. the waves) of the problem consists in permutations of a set of picking points, while the fitness function is the travel distance covered by the picker on the basis of a given sequencing. This means, for instance, that the propagation of a wave cannot be made by applying eq.1, i.e. by adding a random number, as such operation does not make any sense for picking locations and would not generate a usable solution.

On the basis of these (and other similar) considerations, we modified the WWO algorithm by redefining its steps as described below.

Step 1. The starting point of the algorithm is the picking list of items. A random set of $N$ solutions (waves), i.e. permutation of the elements in the picking list, is generated. Each waves initially owns the maximum height $h_{max}$.

Step 2. For each wave $x$, the fitness function $f(x)$ is computed. This reflect the travel distance of a picker who will pick the items in the order they are listed in each sequence. The optimal solution $x^*$ is identified.

Step 3. Waves should be propagated. This procedure is applied to all waves except $x^*$, which is kept unchanged. For the remaining waves, propagation consists in rotating the elements in the picking list, meaning that each element changes its position in the list by an arbitrary, but fixed, number of places, denoted by $\alpha$ ($\alpha \in N^+, 1 \leq \alpha < n$). The new wave is denoted as $x'$. The fitness function of the new wave $f(x')$ should be calculated and compared to the original one $f(x)$. Because we are considering a minimization problem, low value of the fitness functions should be preferred. Therefore, if $f(x) > f(x')$ the new solution will be retained and will replace the original one in the population. Otherwise, the original solution $x$ will be retained, but its height will decrease by 1.

Step 4. For those waves whose height has decreased to 0, breaking is introduced, which means that the wave should disappear from the population and a new one should replace it. For the problem under investigation, the starting point for breaking is the optimal solution $x^*$. A random (but fixed) number of elements $\beta$ ($\beta \in N^+, 1 \leq \beta < n$) of the optimal sequence will be selected and replaced with each other, so as to derive a solution which is different from the optimal one.

Step 5. At any moment in the algorithm running, if the propagation or breaking procedures generate two identical solutions, one of them is retained (with the lowest height), while the second one is replaced with a new wave obtained by breaking.

At each iteration of the algorithm, steps 2-5 are repeated for the whole set of waves in the population. The algorithm typically stops when a given number of iterations $I$ has been carried out. Overall, the parameters of the adapted-WWO algorithm are: $h_{max}$ = the maximum height of a wave; $n$ = number of elements in the picking list; $N$ = size of the population of waves; $\alpha$ = number of positions changed during propagation ($1 < \alpha < n$); $\beta$ = number of positions changed randomly during breaking ($1 < \beta < n$); and $I$ = number of iterations allowed for the algorithm to reach convergence.

To calculate the fitness function, i.e. the travel distance generated by a given sequence of items, the algorithm also needs, as input, the distance between any pair of storage location in the warehouse. Also, the position of the input/output depot should be known, as it is assumed that the picker will start from the depot and return there once he has picked all the items in the picking list.

3. Application example

The adapted-WWO algorithm was implemented under Microsoft Excel® exploiting appropriate macros programmed in visual basic for applications (VBA). Its performance was tested on a 1-block rectangular warehouse, with 12 aisles and 45 storage locations per aisle side, resulting in 1080 storage locations overall. The aisle width is assumed to be negligible compared to the length (narrow aisles), so that its contribution is not included in the computation of the total travel distance. Pickers are assumed to follow an order picking policy, meaning that the picking list corresponds to an order of a final customer. The picker starts from the depot, which is located at the bottom left corner of the warehouse, and returns there once he has picked the whole set of items in the picking list. Low-level picking is assumed.

We consider a picking list composed of $n = 10$ elements. The solution space for a TSP with 10 cities is $10! = 3,628,800$. The size of the initial set of solutions was set at $N = 20$ or $N = 100$ waves. Similarly, $h_{max}$ was set at 2 or 10. The number of iterations was set at $I = 18000$, which should be sufficient for the algorithm to reach convergence. The remaining parameters of the algorithm were set at $\alpha = 2$ and $\beta = 2$. Overall, $2 \times 2 = 4$ configurations of the algorithm setting were considered. For each configuration, the algorithm was launched 1156 times, to get statistically significant results.

As the number of solutions is not excessively high for the problem in exam, an exhaustive procedure has also been implemented in Microsoft Excel® to examine the whole solution space and identify the global optimum. It was found that, for the problem in exam, the shortest path of pickers is 380 m. This result has been used as a benchmark to assess the performance of the adapted-WWO algorithm to identify effective solutions. More precisely, as performance of the algorithm, we derived:
- the capability of the algorithm to reach the optimal solution;
- the number of iterations $NI$ required to reach the optimal solution (mean $\mu_{NI}$ and standard deviation $\sigma_{NI}$ on the whole set of runs);
- the computational time $T$ (mean $\mu_T$ and standard deviation $\sigma_T$ on the whole set of runs) required to reach the optimal solution;
- the computational time per iteration $I_T$.

A first outcome is that the adapted-WWO algorithm is always able to reach the optimal solution for the problem examined, no matter the numerical values set for $N$ and $h_{max}$. With respect to the remaining performance parameters, results are reported in Table 1.

### Table 1: performance of the adapted-WWO algorithm

<table>
<thead>
<tr>
<th>Configuration</th>
<th>$N$</th>
<th>100</th>
<th>20</th>
<th>20</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_{max}$</td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>$\mu_T$ [s]</td>
<td>48</td>
<td>21</td>
<td>70</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>$\sigma_T$ [s]</td>
<td>34</td>
<td>16</td>
<td>72</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>$\mu_{NI}$</td>
<td>194.26</td>
<td>305.91</td>
<td>1014.54</td>
<td>55.63</td>
<td></td>
</tr>
<tr>
<td>$\sigma_{NI}$</td>
<td>138.01</td>
<td>247.45</td>
<td>732.23</td>
<td>44.82</td>
<td></td>
</tr>
<tr>
<td>$I_T$ [s*10^-2]</td>
<td>25</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>37</td>
</tr>
</tbody>
</table>

The results in Table 1 show that, with the increase in $N$, the time required to run the algorithm increases too, which is reasonable, since this generates more solutions to be processed by the algorithm. Also, a lower $h_{max}$ causes the waves to be replaced more frequently, which increases the time required to complete an iteration of the algorithm.

A further result is that the algorithm requires a very low number of iterations ($\mu_{NI} \approx 55$) to reach the optimal solution in the configuration with $N = 100$ and $h_{max} = 2$, while this number increases significantly ($\mu_{NI} \approx 1014$) in the opposite situation ($N = 20$ and $h_{max} = 10$). This seems to suggest that a higher size of the population and lower height of the wave are to be preferred to enhance the computational performance of the algorithm.

A formal design of experiment (DOE) analysis (Montgomery & Runger, 2003) was carried out to assess whether the variables identified above have a significant impact on the performance of the algorithm. Results, in terms of single-factor effects and two-factor interactions, are proposed in Table 2 and 3, respectively for the computational time $T$ and number of iterations $NI$. To assess the significance of the effects observed, the mean square (MS) and the type of effect of the treatment, the mean square of the error (MSE), the significance of the F-test (sig.) have been evaluated. Significant values at $p<0.05$ are highlighted in italic in the Tables.

### Table 2: DOE analysis on the computational time (note: $A=N$, $B=h_{max}$).

<table>
<thead>
<tr>
<th>Effect</th>
<th>A</th>
<th>B</th>
<th>AB</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS</td>
<td>4.99E-07</td>
<td>2.37E-06</td>
<td>2.23E-07</td>
<td>1.15E-07</td>
</tr>
<tr>
<td>MSE</td>
<td>4.33</td>
<td>20.61</td>
<td>2.21</td>
<td></td>
</tr>
<tr>
<td>sig.</td>
<td>0.04</td>
<td>0.00</td>
<td>0.15</td>
<td></td>
</tr>
<tr>
<td>type of effect</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

### Table 3: DOE analysis on the number of iterations (note: $A=N$, $B=h_{max}$).

<table>
<thead>
<tr>
<th>Effect</th>
<th>A</th>
<th>B</th>
<th>AB</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS</td>
<td>2.86E+06</td>
<td>1.02E+06</td>
<td>2.05E+05</td>
<td>1.08E+05</td>
</tr>
<tr>
<td>MSE</td>
<td>26.52</td>
<td>9.42</td>
<td>1.90</td>
<td></td>
</tr>
<tr>
<td>sig.</td>
<td>0.00</td>
<td>0.00</td>
<td>0.18</td>
<td></td>
</tr>
<tr>
<td>type of effect</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

The outcomes in Tables 2 and 3 show that the height of the wave and the population size have a significant effect on both the computational time of the algorithm and the number of iterations required for the algorithm to reach the optimal solution. The effect, however, is different, as the population size exhibits a negative impact on both performance parameters, while the wave height exhibits a positive effect. The interaction between the two parameters never generates a significant effect.

Looking at the significant effects, they should be interpreted as follows. The population size $N$ (factor $A$) has a negative effect on the computational time: therefore, the higher the population size, the lower the computational time required by the algorithm to reach convergence. Obviously, a lower computational time is generally preferred, meaning that we would like to minimise this outcome. Accordingly, a higher $N$ should be set for the adapted-WWO algorithm. Because we tested two values of $N$, i.e. $N = 20$ and $N = 100$ waves, we should conclude that $N = 100$ is the preferred option. Similar considerations can be repeated for the height of the wave, paying attention that, in this case, the impact of $h_{max}$ on the computational time is positive (i.e. a higher $h_{max}$ generates a higher computational time). Therefore, lower values of $h_{max}$ should be preferred. $h_{max} = 2$ thus emerges as the preferred option.

### 4. Conclusions

This paper has proposed a new routing algorithm for the reduction of the travel distance of pickers in manual warehouses. The algorithm grounds on the recent water-wave optimization (WWO) procedure (Zheng, 2015), which has been adapted to make it suitable to be applied to the sequencing problem. The proposed algorithm consists of 5 steps and follows the logic of the WWO as it proposes a “propagation” function and a “breaking” one. The algorithm was coded in Microsoft Excel™, exploiting appropriate VBA macros, which allow the steps of the algorithm to be run automatically.

A test of the algorithm performance was made considering a simple scenario, consisting in a rectangular warehouse...
with 1080 storage locations and a picking list of 10 elements. Two parameters of the algorithm, in particular, were varied in a $2^2$ factorial design, to assess whether they have a significant effect on the results returned.

As the number of elements in the picking list is relatively low, an exhaustive procedure was developed, again in Microsoft Excel™, to explore the whole solution space and identify the global optimal solution. By comparing the results returned by the algorithm with this benchmark, we found that the adapted-WWO procedure is always able to identify the global optimal solution for the problem investigated. The algorithm is also efficient from the computational perspective, as it requires, with the worst setting, 70 s on average to identify the optimal solution. This is fully compatible with the warehouse manager’s need to adopt an algorithm in real-time to route pickers in the warehouse. Also, we have demonstrated, through a formal design of experiments (DOE), that both the population size and the wave height have a significant impact on the algorithm performance, in terms of computational time and number of iterations required to identify the global optimal solution. This suggests that the algorithm can be optimized in its setting, reaching even better performance.

Future research activities are required to test the effect of the remaining parameters (i.e., $\alpha$ and $\beta$) on the results provided by the adapted-WWO algorithm and on the corresponding performance. Also, the performance of the algorithm needs to be compared to that of other routing policies, including “practical” procedures (e.g., S-shaped or largest gap), heuristic ones or nature-inspired algorithms.

References


