Hybrid PACO & Pheromone Initialization for VRPTWs

Wei Shi\textsuperscript{1}, Thomas Weise\textsuperscript{1}, Raymond Chiong\textsuperscript{2}, and Bülent Çatay\textsuperscript{3}

\textsuperscript{1} University of Science and Technology of China, \textsuperscript{2} The University of Newcastle, Australia, \textsuperscript{3} Sabanci University, Turkey

2015-12-10, CIPLS @ SSCI @ Cape Town, South Africa
Introduction: The VRPTW

Vehicle Routing Problem with Time Windows (VRPTW): well-known \textit{NP}-hard distribution logistics problem
Vehicle Routing Problem with Time Windows (VRPTW): well-known NP-hard distribution logistics problem
- homogeneous fleet of $m$ vehicles with capacity $k$
Introduction: The VRPTW

- Vehicle Routing Problem with Time Windows (VRPTW): well-known NP-hard distribution logistics problem
  - homogeneous fleet of $m$ vehicles with capacity $k$ serves $n$ geographically dispersed customers.
Introduction: The VRPTW

Vehicle Routing Problem with Time Windows (VRPTW): well-known \textit{NP}-hard distribution logistics problem

- homogeneous fleet of \( m \) vehicles with capacity \( k \) serves \( n \) geographically dispersed customers.
- customer \( c_i \) has demand \( w_i \)
Introduction: The VRPTW

Vehicle Routing Problem with Time Windows (VRPTW): well-known NP-hard distribution logistics problem

- homogeneous fleet of $m$ vehicles with capacity $k$ serves $n$ geographically dispersed customers.
- customer $c_i$ has demand $w_i$ and service time $s_i$ required for satisfying the demand once a vehicle arrives
Introduction: The VRPTW

- Vehicle Routing Problem with Time Windows (VRPTW): well-known $NP$-hard distribution logistics problem
  - homogeneous fleet of $m$ vehicles with capacity $k$ serves $n$ geographically dispersed customers.
  - customer $c_i$ has demand $w_i$ and service time $s_i$ required for satisfying the demand once a vehicle arrives, which must happen in time window $[e_i, l_i]$
Vehicle Routing Problem with Time Windows (VRPTW): well-known \( NP \)-hard distribution logistics problem

- homogeneous fleet of \( m \) vehicles with capacity \( k \) serves \( n \) geographically dispersed customers.
- customer \( c_i \) has demand \( w_i \) and service time \( s_i \) required for satisfying the demand once a vehicle arrives, which must happen in time window \([e_i, l_i]\)
- all vehicles must leave central depot \( c_0 \) after \( e_0 \) and arrive back before \( l_0 \)
Vehicle Routing Problem with Time Windows (VRPTW): well-known NP-hard distribution logistics problem

- homogeneous fleet of $m$ vehicles with capacity $k$ serves $n$ geographically dispersed customers.
- customer $c_i$ has demand $w_i$ and service time $s_i$ required for satisfying the demand once a vehicle arrives, which must happen in time window $[e_i, l_i]$
- all vehicles must leave central depot $c_0$ after $e_0$ and arrive back before $l_0$
- $d_{i,j}$ is the time required to get to $c_j$ from $c_i$
Vehicle Routing Problem with Time Windows (VRPTW): well-known \textit{NP}-hard distribution logistics problem

- homogeneous fleet of $m$ vehicles with capacity $k$ serves $n$ geographically dispersed customers.
- customer $c_i$ has demand $w_i$ and service time $s_i$ required for satisfying the demand once a vehicle arrives, which must happen in time window $[e_i, l_i]$
- all vehicles must leave central depot $c_0$ after $e_0$ and arrive back before $l_0$
- $d_{i,j}$ is the time required to get to $c_j$ from $c_i$
- each customer must be visited exactly once
Introduction: The VRPTW

Vehicle Routing Problem with Time Windows (VRPTW): well-known NP-hard distribution logistics problem

- homogeneous fleet of $m$ vehicles with capacity $k$ serves $n$ geographically dispersed customers.
- customer $c_i$ has demand $w_i$ and service time $s_i$ required for satisfying the demand once a vehicle arrives, which must happen in time window $[e_i, l_i]$
- all vehicles must leave central depot $c_0$ after $e_0$ and arrive back before $l_0$
- $d_{i,j}$ is the time required to get to $c_j$ from $c_i$
- each customer must be visited exactly once
- vehicle capacity and time windows must not be violated
Introduction: The VRPTW

- Vehicle Routing Problem with Time Windows (VRPTW): well-known $NP$-hard distribution logistics problem
- Two optimization goals
Introduction: The VRPTW

- Vehicle Routing Problem with Time Windows (VRPTW): well-known NP-hard distribution logistics problem
- Two optimization goals:
  \[ f_1: \] number of vehicles needed to serve the customers (minimize)
Introduction: The VRPTW

- Vehicle Routing Problem with Time Windows (VRPTW): well-known NP-hard distribution logistics problem
- Two optimization goals:
  - $f_1$: number of vehicles needed to serve the customers (minimize)
  - $f_2$: total travel distance (minimize)
Introduction: The VRPTW

- Vehicle Routing Problem with Time Windows (VRPTW): well-known \textit{NP}-hard distribution logistics problem
- Two optimization goals:
  - $f_1$: number of vehicles needed to serve the customers (minimize)
  - $f_2$: total travel distance (minimize)
- $f_1$ often considered as more important, since using more vehicles costs more than driving a bit longer
Introduction: The VRPTW

- **Vehicle Routing Problem with Time Windows (VRPTW):** well-known NP-hard distribution logistics problem
- **Two optimization goals**
- **A permutation** \( \pi = (c_i, c_j, \ldots) \) of the cities can be used to encode a solution
Introduction: The VRPTW

- Vehicle Routing Problem with Time Windows (VRPTW): well-known NP-hard distribution logistics problem
- Two optimization goals
- A permutation $\pi = (c_i, c_j, \ldots)$ of the cities can be used to encode a solution:
  - first vehicle leaves depot $c_0$ and travels to $c_i$ servicing it at $b_i = \max\{e_i, e_0 + t_{0i}\}$
Introduction: The VRPTW

- Vehicle Routing Problem with Time Windows (VRPTW): well-known NP-hard distribution logistics problem
- Two optimization goals
- A permutation $\pi = (c_i, c_j, \ldots)$ of the cities can be used to encode a solution:
  - first vehicle leaves depot $c_0$ and travels to $c_i$ servicing it at $b_i = \max \{e_i, e_0 + t_{0i}\}$,
  - then travels to $c_j$, servicing it at $b_j = \max \{e_j, b_i + s_i + t_{ij}\}$.
Vehicle Routing Problem with Time Windows (VRPTW): well-known \textit{NP}-hard distribution logistics problem

Two optimization goals

A permutation $\pi = (c_i, c_j, \ldots)$ of the cities can be used to encode a solution:

- first vehicle leaves depot $c_0$ and travels to $c_i$ servicing it at $b_i = \max \{ e_i, e_0 + t_{0i} \}$,
- then travels to $c_j$, servicing it at $b_j = \max \{ e_j, b_i + s_i + t_{ij} \}$.
- if vehicle capacity is exhausted or no other customer can be visited in time-window restriction, vehicle returns to $c_0$
Introduction: The VRPTW

- Vehicle Routing Problem with Time Windows (VRPTW): well-known NP-hard distribution logistics problem
- Two optimization goals
- A permutation $\pi = (c_i, c_j, \ldots)$ of the cities can be used to encode a solution:
  - first vehicle leaves depot $c_0$ and travels to $c_i$ servicing it at $b_i = \max\{e_i, e_0 + t_{0i}\}$,
  - then travels to $c_j$, servicing it at $b_j = \max\{e_j, b_i + s_i + t_{ij}\}$.
  - if vehicle capacity is exhausted or no other customer can be visited in time-window restriction, vehicle returns to $c_0$
  - next vehicle is used
Vehicle Routing Problem with Time Windows (VRPTW): well-known \textit{NP}-hard distribution logistics problem

Two optimization goals

A permutation $\pi = (c_i, c_j, \ldots)$ of the cities can be used to encode a solution:

- first vehicle leaves depot $c_0$ and travels to $c_i$ servicing it at $b_i = \max\{e_i, e_0 + t_{0i}\}$,
- then travels to $c_j$, servicing it at $b_j = \max\{e_j, b_i + s_i + t_{ij}\}$.
- if vehicle capacity is exhausted or no other customer can be visited in time-window restriction, vehicle returns to $c_0$
- next vehicle is used, until all customers are satisfied
Related Work & Contribution

- Related Work
Related Work:

- exact method: mainly focus on minimizing distance, only feasible on small-scale problems
Related Work & Contribution

- Related Work:
  - exact method: mainly focus on minimizing distance, only feasible on small-scale problems
  - metaheuristics [1]: GAs [2, 3], ACO [4], SA [5, 6], TS [7–9], Adaptive Large Neighborhood Search [10], Variable Neighborhood Search [11], and hybrid methods [12–14]
Related Work & Contribution

Related Work:

- **exact method**: mainly focus on minimizing distance, only feasible on small-scale problems
- **metaheuristics** [1]: GAs [2, 3], ACO [4], SA [5, 6], TS [7–9], Adaptive Large Neighborhood Search [10], Variable Neighborhood Search [11], and hybrid methods [12–14]
- **two-stage approaches** very common [12, 15]: first reduce number of vehicles, then reduce distance
Related Work:

- **exact method**: mainly focus on minimizing distance, only feasible on small-scale problems
- **metaheuristics** [1]: GAs [2, 3], ACO [4], SA [5, 6], TS [7–9], Adaptive Large Neighborhood Search [10], Variable Neighborhood Search [11], and hybrid methods [12–14]
- two-stage approaches very common [12, 15]: first reduce number of vehicles, then reduce distance

**Contribution**
Related Work & Contribution

Related Work:
- exact method: mainly focus on minimizing distance, only feasible on small-scale problems
- metaheuristics \cite{1}: GAs \cite{2, 3}, ACO \cite{4}, SA \cite{5, 6}, TS \cite{7-9}, Adaptive Large Neighborhood Search \cite{10}, Variable Neighborhood Search \cite{11}, and hybrid methods \cite{12-14}
- two-stage approaches very common \cite{12, 15}: first reduce number of vehicles, then reduce distance

Contribution:
- we optimize both objectives at once with a hierarchical approach
Related Work & Contribution

Related Work:

- exact method: mainly focus on minimizing distance, only feasible on small-scale problems
- two-stage approaches very common $[12, 15]$: first reduce number of vehicles, then reduce distance

Contribution:

- we optimize both objectives at once with a hierarchical approach: Solution $\pi_i$ is better than $\pi_2$ if $f_1(\pi_1) < f_1(\pi_2)$ or $f_1(\pi_1) = f_1(\pi_2)$ and $f_2(\pi_1) < f_2(\pi_2)$
Related Work & Contribution

Related Work:
- exact method: mainly focus on minimizing distance, only feasible on small-scale problems
- metaheuristics \cite{1}: GAs \cite{2, 3}, ACO \cite{4}, SA \cite{5, 6}, TS \cite{7–9}, Adaptive Large Neighborhood Search \cite{10}, Variable Neighborhood Search \cite{11}, and hybrid methods \cite{12–14}
- two-stage approaches very common \cite{12, 15}: first reduce number of vehicles, then reduce distance

Contribution:
- we optimize both objectives at once with a hierarchical approach: Solution $\pi_i$ is better than $\pi_2$ if $f_1(\pi_1) < f_1(\pi_2)$ or $f_1(\pi_1) = f_1(\pi_2)$ and $f_2(\pi_1) < f_2(\pi_2)$, which is closest to the nature of the problem.
Related Work & Contribution

Related Work:

- exact method: mainly focus on minimizing distance, only feasible on small-scale problems
- metaheuristics \(^1\): GAs \(^2, 3\), ACO \(^4\), SA \(^5, 6\), TS \(^7–9\), Adaptive Large Neighborhood Search \(^10\), Variable Neighborhood Search \(^11\), and hybrid methods \(^12–14\)
- two-stage approaches very common \(^12, 15\): first reduce number of vehicles, then reduce distance

Contribution:

- we optimize both objectives at once with a hierarchical approach: Solution \(\pi_i\) is better than \(\pi_2\) if \(f_1(\pi_1) < f_1(\pi_2)\) or \(f_1(\pi_1) = f_1(\pi_2)\) and \(f_2(\pi_1) < f_2(\pi_2)\), which is closest to the nature of the problem.
- we investigate the Min-Max Ant System (MMAS) \(^16\), the Ant Colony System (ACS) \(^17\), our previously developed Initialized ACO (IACO) \(^18\), and the Population-based ACO (PACO) algorithm \(^19, 20\)
Related Work & Contribution

Related Work:
- exact method: mainly focus on minimizing distance, only feasible on small-scale problems
- metaheuristics: GAs, ACO, SA, TS, Adaptive Large Neighborhood Search, Variable Neighborhood Search, and hybrid methods
- two-stage approaches very common: first reduce number of vehicles, then reduce distance

Contribution:
- we optimize both objectives at once with a hierarchical approach: Solution $\pi_i$ is better than $\pi_2$ if $f_1(\pi_1) < f_1(\pi_2)$ or $f_1(\pi_1) = f_1(\pi_2)$ and $f_2(\pi_1) < f_2(\pi_2)$, which is closest to the nature of the problem.
- we investigate the Min-Max Ant System (MMAS), the Ant Colony System (ACS), our previously developed Initialized ACO (IACO), and the Population-based ACO (PACO) algorithm.
- we investigate and improve pheromone matrix initialization methods for these methods.
Related Work & Contribution

Related Work:
- exact method: mainly focus on minimizing distance, only feasible on small-scale problems
- metaheuristics \(^{[1]}\): GAs \(^{[2, 3]}\), ACO \(^{[4]}\), SA \(^{[5, 6]}\), TS \(^{[7–9]}\), Adaptive Large Neighborhood Search \(^{[10]}\), Variable Neighborhood Search \(^{[11]}\), and hybrid methods \(^{[12–14]}\)
- two-stage approaches very common \(^{[12, 15]}\): first reduce number of vehicles, then reduce distance

Contribution:
- we optimize both objectives at once with a hierarchical approach: Solution \(\pi_i\) is better than \(\pi_2\) if \(f_1(\pi_1) < f_1(\pi_2)\) or \(f_1(\pi_1) = f_1(\pi_2)\) and \(f_2(\pi_1) < f_2(\pi_2)\), which is closest to the nature of the problem.
- we investigate the Min-Max Ant System (MMAS) \(^{[16]}\), the Ant Colony System (ACS) \(^{[17]}\), our previously developed Initialized ACO (IACO) \(^{[18]}\), and the Population-based ACO (PACO) algorithm \(^{[19, 20]}\)
- we investigate and improve pheromone matrix initialization methods for these methods
- we hybridize the algorithm to further improve the result quality
Existing ACO Methods

- Solomon benchmark set \cite{21}: 25-, 50-, and 100-customer instance sets
Existing ACO Methods

- Solomon benchmark set [21]: 25-, 50-, and 100-customer instance sets
- 20 independent runs per instance, 300’000 FEs per run
Existing ACO Methods

- Solomon benchmark set \cite{21}: 25-, 50-, and 100-customer instance sets
- 20 independent runs per instance, 300'000 FEs per run

<table>
<thead>
<tr>
<th>Algorithm 1 vs. 2</th>
<th>Instances with 25 customers</th>
<th>Instances with 50 customers</th>
<th>Instances with 100 customers</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$f_1$</td>
<td>$f_2$</td>
<td>$f_1$</td>
<td>$f_2$</td>
</tr>
<tr>
<td>ACS IACO</td>
<td>- + 0</td>
<td>- + 0</td>
<td>- + 0</td>
<td>- + 0</td>
</tr>
<tr>
<td>ACS MMAS</td>
<td>0 40 16</td>
<td>0 52 4</td>
<td>0 40 16</td>
<td>1 51 4</td>
</tr>
<tr>
<td>ACS PACO-ABS</td>
<td>0 49 7</td>
<td>1 51 4</td>
<td>0 41 15</td>
<td>9 36 11</td>
</tr>
<tr>
<td>ACS PACO-EBS</td>
<td>0 12 44</td>
<td>9 31 16</td>
<td>0 35 21</td>
<td>5 45 6</td>
</tr>
<tr>
<td>ACS PACO-QBS</td>
<td>0 47 9</td>
<td>0 54 2</td>
<td>0 48 8</td>
<td>0 54 2</td>
</tr>
<tr>
<td>IACO MMAS</td>
<td>0 17 39 13 28 15</td>
<td>9 7 40 38 7</td>
<td>11</td>
<td>35 1 20 56 0</td>
</tr>
<tr>
<td>IACO PACO-ABS</td>
<td>0 12 44</td>
<td>9 31 16</td>
<td>0 35 21</td>
<td>5 45 6</td>
</tr>
<tr>
<td>IACO PACO-EBS</td>
<td>0 9 47 9 28 19</td>
<td>0 31 25</td>
<td>4 43 9</td>
<td>0 42 14</td>
</tr>
<tr>
<td>IACO PACO-QBS</td>
<td>0 13 43 8 34 14</td>
<td>0 37 19</td>
<td>4 49 3</td>
<td>0 54 2</td>
</tr>
<tr>
<td>MMAS PACO-ABS</td>
<td>5 1 50 15 12 29</td>
<td>0 31 25</td>
<td>1 49 6</td>
<td>0 54 2</td>
</tr>
<tr>
<td>MMAS PACO-EBS</td>
<td>3 1 52 19 12 25</td>
<td>0 29 27</td>
<td>0 49 7</td>
<td>0 53 3</td>
</tr>
<tr>
<td>MMAS PACO-QBS</td>
<td>3 1 52 16 15 25</td>
<td>0 33 23</td>
<td>0 52 4</td>
<td>0 55 1</td>
</tr>
<tr>
<td>PACO-ABS PACO-EBS</td>
<td>0 56 5 1 50 3</td>
<td>0 53 19 2</td>
<td>35 28 0 28 48 0</td>
<td>8 31 0 137 72 3 93</td>
</tr>
<tr>
<td>PACOABS PACO-QBS</td>
<td>0 56 0 3 53 0</td>
<td>1 55 0 13 43 0 1 55 1 28 27</td>
<td>0 2 166 1 44 123</td>
<td></td>
</tr>
<tr>
<td>PACO-ABS PACO-QBS</td>
<td>1 55 0 18 38 0 4 52 0 34 22 0 39 17 0 55 1</td>
<td>1 43 124 0 107 61</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Mann-Whitey U test ($\alpha = 0.02$) comparison results for ACO algorithms (− is better, + is worse).
Existing ACO Methods

- Solomon benchmark set [21]: 25-, 50-, and 100-customer instance sets
- 20 independent runs per instance, 300’000 FE's per run

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Instances with 25 customers</th>
<th>Instances with 50 customers</th>
<th>Instances with 100 customers</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACS</td>
<td>f₁ +</td>
<td>f₂</td>
<td>f₁ +</td>
<td>f₂</td>
</tr>
<tr>
<td>IACO</td>
<td>0 40 16</td>
<td>0 52 4</td>
<td>0 40 16</td>
<td>1 51 4</td>
</tr>
<tr>
<td>MMAS</td>
<td>0 49 7</td>
<td>1 51 4</td>
<td>0 41 15</td>
<td>9 36 11</td>
</tr>
<tr>
<td>PACO-ABS</td>
<td>0 12 44</td>
<td>9 31 16</td>
<td>0 35 21</td>
<td>5 45 6</td>
</tr>
<tr>
<td>PACO-EBS</td>
<td>0 47 9</td>
<td>0 54 2</td>
<td>0 48 8</td>
<td>0 54 2</td>
</tr>
<tr>
<td>PACO-QBS</td>
<td>0 45 11</td>
<td>0 55 1</td>
<td>0 49 7</td>
<td>0 55 1</td>
</tr>
<tr>
<td>ACS</td>
<td>IACO</td>
<td>MMAS</td>
<td>PACO-ABS</td>
<td>PACO-EBS</td>
</tr>
<tr>
<td>IACO</td>
<td>0 17 39</td>
<td>13 28 15</td>
<td>9 7 40</td>
<td>38 7</td>
</tr>
<tr>
<td>IACO</td>
<td>0 12 44</td>
<td>9 31 16</td>
<td>0 35 21</td>
<td>5 45 6</td>
</tr>
<tr>
<td>IACO</td>
<td>0 13 43</td>
<td>8 34 14</td>
<td>0 37 19</td>
<td>4 49 3</td>
</tr>
<tr>
<td>MMAS</td>
<td>5 1 50</td>
<td>15 12 29</td>
<td>0 31 25</td>
<td>1 49 6</td>
</tr>
<tr>
<td>MMAS</td>
<td>3 1 52</td>
<td>19 12 25</td>
<td>0 29 27</td>
<td>0 49 7</td>
</tr>
<tr>
<td>MMAS</td>
<td>3 1 52</td>
<td>16 15 25</td>
<td>0 33 23</td>
<td>0 52 4</td>
</tr>
<tr>
<td>PACO-ABS</td>
<td>0 0 56</td>
<td>5 1 50</td>
<td>3 0 53</td>
<td>19 2</td>
</tr>
<tr>
<td>PACO-QBS</td>
<td>0 0 56</td>
<td>0 3 53</td>
<td>0 1 55</td>
<td>0 13 43</td>
</tr>
<tr>
<td>PACO-EBS</td>
<td>1 0 55</td>
<td>0 18 38</td>
<td>0 4 52</td>
<td>0 34 22</td>
</tr>
</tbody>
</table>

Mann-Whitey U test ($\alpha = 0.02$) comparison results for ACO algorithms (− is better, + is worse).

- ACS performs worst
Existing ACO Methods

- Solomon benchmark set \[^{[21]}\]: 25-, 50-, and 100-customer instance sets
- 20 independent runs per instance, 300,000 FEs per run

### Algorithm 1 vs. 2

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Instances with 25 customers</th>
<th>Instances with 50 customers</th>
<th>Instances with 100 customers</th>
<th>Total</th>
</tr>
</thead>
</table>
| ACS       | \[\begin{array}{c}
  \text{f}_1 \\
  \text{f}_2 \\
\end{array}\] | \[\begin{array}{c}
  \text{f}_1 \\
  \text{f}_2 \\
\end{array}\] | \[\begin{array}{c}
  \text{f}_1 \\
  \text{f}_2 \\
\end{array}\] | \[\begin{array}{c}
  \text{f}_1 \\
  \text{f}_2 \\
\end{array}\] |
| IACO      | 0 40 16 0 52 4              | 0 40 16 1 51 4              | 0 44 12 0 54 2              | 0 124 44 1 157 10 |
| MMAS      | 0 49 7 1 51 4              | 0 41 15 9 36 11             | 2 36 18 35 7 14             | 2 126 40 45 94 29 |
| PACO-ABS  | 0 12 44 9 31 16             | 0 35 21 5 45 6              | 0 54 2 0 51 5              | 0 101 67 14 127 27 |
| PACO-EBS  | 0 47 9 0 54 2              | 0 48 8 0 54 2              | 0 55 1 0 52 4              | 0 150 18 0 160 8 |
| PACO-QBS  | 0 45 11 0 55 1              | 0 49 7 0 55 1              | 0 56 0 0 56 0              | 0 150 18 0 166 2 |
| IACO      | 0 17 39 13 28 15            | 9 7 40 38 7 11             | 35 1 20 56 0 0             | 44 25 99 107 35 26 |
| PACO-ABS  | 0 12 44 9 31 16             | 0 35 21 5 45 6              | 0 54 2 0 51 5              | 0 101 67 14 127 27 |
| PACO-EBS  | 0 9 47 9 28 19              | 0 31 25 4 43 9              | 0 42 14 2 39 15             | 0 82 86 15 110 43 |
| PACO-QBS  | 0 13 43 8 34 14              | 0 37 19 4 49 3              | 0 54 2 0 52 4              | 0 104 64 12 135 21 |
| MMAS      | 0 5 50 15 12 29              | 0 31 25 1 49 6              | 0 54 2 0 56 0              | 5 86 77 16 117 35 |
| MMAS      | 3 1 52 19 12 25              | 0 29 27 0 49 7              | 0 53 3 0 56 0              | 3 89 82 19 117 32 |
| MMAS      | 3 1 52 16 15 25              | 0 33 23 0 52 4              | 0 55 1 0 56 0              | 3 83 76 16 123 29 |
| PACO-ABS  | 0 0 56 5 1 50               | 0 53 19 2 35 28            | 0 48 0 8                   | 31 0 137 72 3 93 |
| PACO-QBS  | 0 0 56 0 3 53               | 0 1 55 0 13 43             | 0 1 55 1 28 27             | 0 2 166 1 44 123 |
| PACO-EBS  | 1 0 55 0 18 38              | 0 4 52 0 34 22             | 0 39 17 0 55 1             | 1 43 124 0 107 61 |

Mann-Whitey U test (\(\alpha = 0.02\)) comparison results for ACO algorithms (− is better, + is worse).

- ACS performs worst
- PACO with QBS rule performs best
Existing ACO Methods

- Solomon benchmark set [21]: 25-, 50-, and 100-customer instance sets
- 20 independent runs per instance, 300,000 FEs per run

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Instances with 25 customers</th>
<th>Instances with 50 customers</th>
<th>Instances with 100 customers</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$f_1$</td>
<td>$f_2$</td>
<td>$f_1$</td>
<td>$f_2$</td>
</tr>
<tr>
<td>ACS</td>
<td>0</td>
<td>40</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>IACO</td>
<td>0</td>
<td>49</td>
<td>1</td>
<td>51</td>
</tr>
<tr>
<td>MMAS</td>
<td>0</td>
<td>12</td>
<td>44</td>
<td>9</td>
</tr>
<tr>
<td>PACO-ABS</td>
<td>0</td>
<td>47</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>PACO-EBS</td>
<td>0</td>
<td>45</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>ACS</td>
<td>0</td>
<td>17</td>
<td>39</td>
<td>13</td>
</tr>
<tr>
<td>IACO</td>
<td>0</td>
<td>12</td>
<td>44</td>
<td>9</td>
</tr>
<tr>
<td>PACO-EBS</td>
<td>0</td>
<td>9</td>
<td>47</td>
<td>9</td>
</tr>
<tr>
<td>ACS</td>
<td>0</td>
<td>13</td>
<td>43</td>
<td>8</td>
</tr>
<tr>
<td>MMAS</td>
<td>5</td>
<td>1</td>
<td>50</td>
<td>15</td>
</tr>
<tr>
<td>MMAS</td>
<td>3</td>
<td>1</td>
<td>52</td>
<td>19</td>
</tr>
<tr>
<td>MMAS</td>
<td>3</td>
<td>1</td>
<td>52</td>
<td>16</td>
</tr>
<tr>
<td>PACO-ABS</td>
<td>0</td>
<td>0</td>
<td>56</td>
<td>5</td>
</tr>
<tr>
<td>PACO-QBS</td>
<td>0</td>
<td>0</td>
<td>56</td>
<td>0</td>
</tr>
<tr>
<td>PACO-EBS</td>
<td>1</td>
<td>0</td>
<td>55</td>
<td>0</td>
</tr>
</tbody>
</table>

Mann-Whitey U test ($\alpha = 0.02$) comparison results for ACO algorithms ($-$ is better, $+$ is worse).

- ACS performs worst
- PACO with QBS rule performs best $\Rightarrow$ use from now on
Pheromone Initialization

- Utilize static information from problem instance to initialize pheromones for PACO-QBS
Pheromone Initialization

- Utilize static information from problem instance to initialize pheromones for PACO-QBS
- Model service begin time $b_i$ as random variable $PD$
Pheromone Initialization

- Utilize static information from problem instance to initialize pheromones for PACO-QBS
- Model service begin time $b_i$ as random variable $PD$ (for $PD$, we test normal, uniform, and power distribution PDFs)
Pheromone Initialization

- Utilize static information from problem instance to initialize pheromones for PACO-QBS
- Model service begin time $b_i$ as random variable $PD$
- Define $VE$ as a function which is larger if $c_i$ and $c_j$ are close and if $c_j$ would be serviced at the end of its time window if visited directly after $c_i$
Pheromone Initialization

- Utilize static information from problem instance to initialize pheromones for PACO-QBS
- Model service begin time $b_i$ as random variable $PD$
- Define $VE$ as a function which is larger if $c_i$ and $c_j$ are close
- Set $\tau_{i,j}^{0} \approx \max \left\{ \frac{1}{n}, \int_{e_i}^{e_j} PD(x) \ast VE(i, j, x) dx \right\}$
Pheromone Initialization

- Utilize static information from problem instance to initialize pheromones for PACO-QBS
- Model service begin time $b_i$ as random variable $PD$
- Define $VE$ as a function which is larger if $c_i$ and $c_j$ are close
- Set $\tau_{i,j}^0 \approx \max \left\{ \frac{1}{n}, \int_{e_i}^{l_i} PD(x) \ast VE(i, j, x) dx \right\}$
- Experiments with PACO-QBS and the three different probability distribution models show...
Pheromone Initialization

- Utilize static information from problem instance to initialize pheromones for PACO-QBS
- Model service begin time $b_i$ as random variable $PD$
- Define $VE$ as a function which is larger if $c_i$ and $c_j$ are close
- Set $\tau_{i,j}^0 \approx \max \left\{ \frac{1}{n}, \int_{e_i}^{l_i} PD(x) \ast VE(i, j, x) dx \right\}$
- Experiments with PACO-QBS and the three different probability distribution models show...

<table>
<thead>
<tr>
<th></th>
<th>Instances with 25 customers</th>
<th>Instances with 50 customers</th>
<th>Instances with 100 customers</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$f_1$</td>
<td>$f_2$</td>
<td>$f_1$</td>
<td>$f_2$</td>
</tr>
<tr>
<td>Algorithm 1 vs. 2</td>
<td>-</td>
<td>+</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>Nolni Normal</td>
<td>0</td>
<td>2</td>
<td>54</td>
<td>4</td>
</tr>
<tr>
<td>Nolni Power</td>
<td>0</td>
<td>3</td>
<td>53</td>
<td>1</td>
</tr>
<tr>
<td>Nolni Uniform</td>
<td>0</td>
<td>2</td>
<td>54</td>
<td>1</td>
</tr>
<tr>
<td>Normal Power</td>
<td>0</td>
<td>0</td>
<td>56</td>
<td>1</td>
</tr>
<tr>
<td>Normal Uniform</td>
<td>0</td>
<td>0</td>
<td>56</td>
<td>0</td>
</tr>
<tr>
<td>Power Uniform</td>
<td>0</td>
<td>0</td>
<td>56</td>
<td>4</td>
</tr>
</tbody>
</table>
Pheromone Initialization

- Utilize static information from problem instance to initialize pheromones for PACO-QBS
- Model service begin time $b_i$ as random variable $PD$
- Define $VE$ as a function which is larger if $c_i$ and $c_j$ are close
- Set $\tau_{i,j}^0 \approx \max\left\{ \frac{1}{n}, \int_{e_i}^{l_i} PD(x) \ast VE(i, j, x)dx \right\}$
- Experiments with PACO-QBS and the three different probability distribution models show that pheromone-initialized PACO performs significantly better

<table>
<thead>
<tr>
<th>Instances with 25 customers</th>
<th>Instances with 50 customers</th>
<th>Instances with 100 customers</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$ $f_2$</td>
<td>$f_1$ $f_2$</td>
<td>$f_1$ $f_2$</td>
<td>$f_1$ $f_2$</td>
</tr>
<tr>
<td>Algorithm 1 vs. 2</td>
<td>-</td>
<td>+</td>
<td>0</td>
</tr>
<tr>
<td>NoIni Normal</td>
<td>0</td>
<td>2</td>
<td>54</td>
</tr>
<tr>
<td>NoIni Power</td>
<td>0</td>
<td>3</td>
<td>53</td>
</tr>
<tr>
<td>NoIni Uniform</td>
<td>0</td>
<td>2</td>
<td>54</td>
</tr>
<tr>
<td>Normal Power</td>
<td>0</td>
<td>0</td>
<td>56</td>
</tr>
<tr>
<td>Normal Uniform</td>
<td>0</td>
<td>0</td>
<td>56</td>
</tr>
<tr>
<td>Power Uniform</td>
<td>0</td>
<td>0</td>
<td>56</td>
</tr>
</tbody>
</table>
Pheromone Initialization

- Utilize static information from problem instance to initialize pheromones for PACO-QBS
- Model service begin time \( b_i \) as random variable \( PD \)
- Define \( VE \) as a function which is larger if \( c_i \) and \( c_j \) are close
- Set \( \tau_{i,j}^0 \approx \max \left\{ \frac{1}{n}, \int_{e_i}^{e_j} PD(x) * VE(i,j,x) dx \right\} \)
- Experiments with PACO-QBS and the three different probability distribution models show that pheromone-initialized PACO performs significantly better and power distributed \( b \) performs best

<table>
<thead>
<tr>
<th></th>
<th>Instances with 25 customers</th>
<th>Instances with 50 customers</th>
<th>Instances with 100 customers</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( f_1 )</td>
<td>( f_2 )</td>
<td>( f_1 )</td>
<td>( f_2 )</td>
</tr>
<tr>
<td>Algorithm 1 vs. 2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Nolni Normal</td>
<td>0</td>
<td>2</td>
<td>54</td>
<td>4</td>
</tr>
<tr>
<td>Nolni Power</td>
<td>0</td>
<td>3</td>
<td>53</td>
<td>1</td>
</tr>
<tr>
<td>Nolni Uniform</td>
<td>0</td>
<td>2</td>
<td>54</td>
<td>1</td>
</tr>
<tr>
<td>Normal Power</td>
<td>0</td>
<td>0</td>
<td>56</td>
<td>1</td>
</tr>
<tr>
<td>Normal Uniform</td>
<td>0</td>
<td>0</td>
<td>56</td>
<td>0</td>
</tr>
<tr>
<td>Power Uniform</td>
<td>0</td>
<td>0</td>
<td>56</td>
<td>4</td>
</tr>
</tbody>
</table>
Ideally, initialization should assign pheromones such that the edges with the strongest pheromones form larger tour components.
Improved Initialization

- Ideally, initialization should assign pheromones such that the edges with the strongest pheromones form larger tour components.
- This works especially for instances where customers are clustered.
**Improved Initialization**

- Ideally, initialization should assign pheromones such that the edges with the strongest pheromones form larger tour components.
- This works especially for instances where customers are clustered, but not if customers and time windows are completely random.
Improved Initialization

- Ideally, initialization should assign pheromones such that the edges with the strongest pheromones form larger tour components.
- This works especially for instances where customers are clustered, but not if customers and time windows are completely random.
- Method 1: Change $VE$ to put more pheromones on shorter edges.
Improved Initialization

- Ideally, initialization should assign pheromones such that the edges with the strongest pheromones form larger tour components.
- This works especially for instances where customers are clustered, but not if customers and time windows are completely random.
- Method 1: Change VE to put more pheromones on shorter edges.
- Method 2: Keep initialized pheromone only on one edge per node; two choices maximum or difference selection (see paper).
Improved Initialization

- Ideally, initialization should assign pheromones such that the edges with the strongest pheromones form larger tour components.
- This works especially for instances where customers are clustered, but not if customers and time windows are completely random.
- Method 1: Change $VE$ to put more pheromones on shorter edges.
- Method 2: Keep initialized pheromone only on one edge per node; two choices maximum or difference selection (see paper).
Improved Initialization

- Ideally, initialization should assign pheromones such that the edges with the strongest pheromones form larger tour components.
- This works especially for instances where customers are clustered, but not if customers and time windows are completely random.
- Method 1: Change VE to put more pheromones on shorter edges.
- Method 2: Keep initialized pheromone only on one edge per node; two choices maximum or difference selection (see paper).
Improved Initialization

- Ideally, initialization should assign pheromones such that the edges with the strongest pheromones form larger tour components.
- This works especially for instances where customers are clustered, but not if customers and time windows are completely random.
- Method 1: Change VE to put more pheromones on shorter edges.
- Method 2: Keep initialized pheromone only on one edge per node; two choices: maximum or difference selection (see paper).
Homberger and Gehring [12] proposed a hybrid metaheuristic that randomly selects one neighborhood from \( \{ N_{1-insert}, N_{1-exchange}, N_{2-opt} \} \) to refine solutions with local search.
Homberger and Gehring \cite{12} proposed a hybrid metaheuristic that randomly selects one neighborhood from \{\(N_1\text{-}insert\), \(N_1\text{-}exchange\), \(N_2\text{-}opt\}\) to refine solutions with local search.
Homberger and Gehring [12] proposed a hybrid metaheuristic that randomly selects one neighborhood from 
\{N_{1-insert}, N_{1-exchange}, N_{2-opt}\} to refine solutions with local search.

We adopt this mechanism into PI-PACO.
Hybridize with Local Search

- Homberger and Gehring [12] proposed a hybrid metaheuristic that randomly selects one neighborhood from \( \{ N_{\text{insert}}, N_{\text{exchange}}, N_{\text{opt}} \} \) to refine solutions with local search.
- We adopt this mechanism into PI-PACO.
- Hybrid PI-PACO with difference selection achieves better results than hybrid PACO without pheromone initialization.

<table>
<thead>
<tr>
<th>Type</th>
<th>Goal</th>
<th>Chen and Ting [14]</th>
<th>Sodsoon and Changyom [22]</th>
<th>hybrid PACO</th>
<th>hybrid PI-PACO maximum selection</th>
<th>hybrid PI-PACO difference selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>( f_1 )</td>
<td>12.83</td>
<td>13.83</td>
<td>12.83</td>
<td>12.92</td>
<td>12.75</td>
</tr>
<tr>
<td></td>
<td>( f_2 )</td>
<td>1203.56</td>
<td>1259.19</td>
<td>1204.06</td>
<td>1205.11</td>
<td>1203.67</td>
</tr>
<tr>
<td>C1</td>
<td>( f_1 )</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>( f_2 )</td>
<td>828.76</td>
<td>838.12</td>
<td>828.61</td>
<td>828.60</td>
<td>828.55</td>
</tr>
<tr>
<td>RC1</td>
<td>( f_1 )</td>
<td>12.50</td>
<td>12.63</td>
<td>12.75</td>
<td>12.63</td>
<td>12.38</td>
</tr>
<tr>
<td></td>
<td>( f_2 )</td>
<td>1363.84</td>
<td>1436.58</td>
<td>1381.42</td>
<td>1380.78</td>
<td>1380.54</td>
</tr>
<tr>
<td>R2</td>
<td>( f_1 )</td>
<td>3.09</td>
<td>3.82</td>
<td>3.45</td>
<td>3.64</td>
<td>3.54</td>
</tr>
<tr>
<td></td>
<td>( f_2 )</td>
<td>932.23</td>
<td>980.98</td>
<td>1005.35</td>
<td>995.03</td>
<td>1006.38</td>
</tr>
<tr>
<td>C2</td>
<td>( f_1 )</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>( f_2 )</td>
<td>589.86</td>
<td>591.13</td>
<td>590.71</td>
<td>589.93</td>
<td>589.86</td>
</tr>
<tr>
<td>RC2</td>
<td>( f_1 )</td>
<td>3.75</td>
<td>4.5</td>
<td>4.13</td>
<td>4.38</td>
<td>4.13</td>
</tr>
<tr>
<td></td>
<td>( f_2 )</td>
<td>1079.81</td>
<td>1141.63</td>
<td>1113.59</td>
<td>1156.20</td>
<td>1109.8</td>
</tr>
</tbody>
</table>
Hybridize with Local Search

- We adopt this mechanism into PI-PACO.
- Hybrid PI-PACO with difference selection achieves better results than hybrid PACO without pheromone initialization.
- It outperforms the hybrid algorithm by Chen and Ting [14] on problem type C1 and achieves similar results on problem type C2.

<table>
<thead>
<tr>
<th>Type</th>
<th>Goal</th>
<th>Chen and Ting [14]</th>
<th>Sodsoon and Changyom [22]</th>
<th>hybrid PACO</th>
<th>hybrid PI-PACO maximum selection</th>
<th>hybrid PI-PACO difference selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>$f_1$</td>
<td>12.83</td>
<td>13.83</td>
<td>12.83</td>
<td>12.92</td>
<td>12.75</td>
</tr>
<tr>
<td></td>
<td>$f_2$</td>
<td>1203.56</td>
<td>1259.19</td>
<td>1204.06</td>
<td>1205.11</td>
<td>1203.67</td>
</tr>
<tr>
<td>C1</td>
<td>$f_1$</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>$f_2$</td>
<td>828.76</td>
<td>838.12</td>
<td>828.61</td>
<td>828.60</td>
<td>828.55</td>
</tr>
<tr>
<td>RC1</td>
<td>$f_1$</td>
<td>12.50</td>
<td>12.63</td>
<td>12.75</td>
<td>12.63</td>
<td>12.38</td>
</tr>
<tr>
<td></td>
<td>$f_2$</td>
<td>1363.84</td>
<td>1436.58</td>
<td>1381.42</td>
<td>1380.78</td>
<td>1380.54</td>
</tr>
<tr>
<td>R2</td>
<td>$f_1$</td>
<td>3.09</td>
<td>3.82</td>
<td>3.45</td>
<td>3.64</td>
<td>3.54</td>
</tr>
<tr>
<td></td>
<td>$f_2$</td>
<td>932.23</td>
<td>980.98</td>
<td>1005.35</td>
<td>995.03</td>
<td>1006.38</td>
</tr>
<tr>
<td>C2</td>
<td>$f_1$</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>$f_2$</td>
<td>589.86</td>
<td>591.13</td>
<td>590.71</td>
<td>589.93</td>
<td>589.86</td>
</tr>
<tr>
<td>RC2</td>
<td>$f_1$</td>
<td>3.75</td>
<td>4.5</td>
<td>4.13</td>
<td>4.38</td>
<td>4.13</td>
</tr>
<tr>
<td></td>
<td>$f_2$</td>
<td>1079.81</td>
<td>1141.63</td>
<td>1113.59</td>
<td>1156.20</td>
<td>1109.8</td>
</tr>
</tbody>
</table>
Hybridize with Local Search

- We adopt this mechanism into PI-PACO.
- Hybrid PI-PACO with difference selection achieves better results than hybrid PACO without pheromone initialization.
- It is similar to the MMAS-VRPTW [22] but outperforms it on all except R2 instances in terms of the distance.

<table>
<thead>
<tr>
<th>Type</th>
<th>Goal</th>
<th>Chen and Ting [14]</th>
<th>Sodsoon and Changyom [22]</th>
<th>hybrid PACO</th>
<th>hybrid PI-PACO maximum selection</th>
<th>hybrid PI-PACO difference selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>$f_1$</td>
<td>12.83</td>
<td>13.83</td>
<td>12.83</td>
<td>12.92</td>
<td>12.75</td>
</tr>
<tr>
<td></td>
<td>$f_2$</td>
<td>1203.56</td>
<td>1259.19</td>
<td>1204.06</td>
<td>1205.11</td>
<td>1203.67</td>
</tr>
<tr>
<td>C1</td>
<td>$f_1$</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>$f_2$</td>
<td>828.76</td>
<td>838.12</td>
<td>828.61</td>
<td>828.60</td>
<td>828.55</td>
</tr>
<tr>
<td>RC1</td>
<td>$f_1$</td>
<td>12.50</td>
<td>12.63</td>
<td>12.75</td>
<td>12.63</td>
<td>12.38</td>
</tr>
<tr>
<td></td>
<td>$f_2$</td>
<td>1363.84</td>
<td>1436.58</td>
<td>1381.42</td>
<td>1380.78</td>
<td>1380.54</td>
</tr>
<tr>
<td>R2</td>
<td>$f_1$</td>
<td>3.09</td>
<td>3.82</td>
<td>3.45</td>
<td>3.64</td>
<td>3.54</td>
</tr>
<tr>
<td></td>
<td>$f_2$</td>
<td>932.23</td>
<td>980.98</td>
<td>1005.35</td>
<td>995.03</td>
<td>1006.38</td>
</tr>
<tr>
<td>C2</td>
<td>$f_1$</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>$f_2$</td>
<td>589.86</td>
<td>591.13</td>
<td>590.71</td>
<td>589.93</td>
<td>589.86</td>
</tr>
<tr>
<td>RC2</td>
<td>$f_1$</td>
<td>3.75</td>
<td>4.5</td>
<td>4.13</td>
<td>4.38</td>
<td>4.13</td>
</tr>
<tr>
<td></td>
<td>$f_2$</td>
<td>1079.81</td>
<td>1141.63</td>
<td>1113.59</td>
<td>1156.20</td>
<td>1109.8</td>
</tr>
</tbody>
</table>
PACO best ACO for VRPTW
Summary

- PACO best ACO for VRPTW
- Pheromone matrix initialization makes it better
Summary

- PACO best ACO for VRPTW
- Pheromone matrix initialization makes it better
- Hybridization + pheromone matrix initialization is best
Summary

- PACO best ACO for VRPTW
- Pheromone matrix initialization makes it better
- Hybridization + pheromone matrix initialization is best
- Concept should be tested in other domains, such as quadratic assignment problems
谢谢！

Thank you.

Wei Shi¹, Thomas Weise¹, Raymond Chiong², and Bülent Çatay³
¹ University of Science and Technology of China,
² The University of Newcastle, Australia
³ Sabanci University, Turkey


