Fast High-Quality Tabletop Rearrangement in Bounded Workspace

Kai Gao¹  Darren Lau²  Baichuan Huang¹
Kostas E. Bekris¹  Jingjin Yu¹

¹Computer Science @ Rutgers University
²Computer Science @ Cornell University

Submitted to 2022 International Conference on Robotics and Automation (ICRA 2022)
Contents

- Problem
  - Tabletop rearrangement problem
  - Constraints and object dependencies

- Algorithmic solutions
  - Local planner
  - High level framework
  - Preprocessing routine

- Experimental results
  - Quantitative analysis
  - Hardware demonstration
Tabletop Object Rearrangement

Start Arrangement

Goal Arrangement
Tabletop Object Rearrangement

Start Arrangement

Goal Arrangement

Dependency Graph
Lazy Buffer Allocation

Instance

Primitive Plan
1. Move $O_1$ to a buffer
2. Move $O_2$ to the goal
3. Move $O_1$ to the goal
4. Move $O_3$ to the goal

Rearrangement Plan
1. Move $O_1$ to (4.6, 1.1)
2. Move $O_2$ to (1.2, 2.7)
3. Move $O_1$ to (2.2, 1.1)
4. Move $O_3$ to (4.5, 2.0)

Primitive Planning

Buffer Allocation
Lazy Buffer Allocation

Instance

Primitive Plan
1. Move $O_1$ to a buffer
2. Move $O_2$ to the goal
3. Move $O_1$ to the goal
4. Move $O_3$ to the goal

Rearrangement Plan
1. Move $O_1$ to (4.6, 1.1)
2. Move $O_2$ to (1.2, 2.7)
3. Move $O_1$ to (2.2, 1.1)
4. Move $O_3$ to (4.5, 2.0)
Tabletop Rearrangement with Lazy Buffer Allocation

Primitive Plan
1. Move $O_1$ to a buffer
2. Move $O_3$ to a buffer
3. Move $O_2$ to the goal
4. Move $O_1$ to the goal
5. Move $O_3$ to the goal

Rearrangement Plan
1. Move $O_1$ to (3.7, 2.5)
2. Move $O_3$ to (4.7, 1.8)
3. Move $O_2$ to (1.3, 3.5)
4. Move $O_1$ to (1.3, 2.2)
5. Move $O_3$ to (2.5, 2.3)
Tabletop Rearrangement with Lazy Buffer Allocation

Rearrangement Plan
1. Move $O_1$ to (3.7, 2.5)
2. Move $O_3$ to (4.7, 1.8)
3. Move $O_2$ to (1.3, 3.5)
4. Move $O_2$ to (1.3, 2.2)
5. Move $O_3$ to (2.5, 2.3)

By accepting partial plans, success rate increases significantly in hard cases.
Solvers with the preprocessing is around 100 times faster at the price of 30% more pick-n-place actions.
Comparison with other Methods

### Computation Time

<table>
<thead>
<tr>
<th>Method</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBM-SP-BST</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>MCTS</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>BiRRT(fmRS)</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>11</td>
</tr>
</tbody>
</table>

### # Collision Checks

<table>
<thead>
<tr>
<th># Objects</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBM-SP-BST</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>MCTS</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>BiRRT(fmRS)</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>11</td>
</tr>
</tbody>
</table>

### # actions(*#objects)

<table>
<thead>
<tr>
<th># Objects</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBM-SP-BST</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>MCTS</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>BiRRT(fmRS)</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>11</td>
</tr>
</tbody>
</table>

### Success Rate

- **RBM-SP-BST**
- **MCTS**
- **BiRRT(fmRS)**

<table>
<thead>
<tr>
<th># Objects</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>24</th>
<th>35</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBM-SP-BST</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>MCTS</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
</tbody>
</table>
Comparative Study: Cylindrical Objects

TRLB (ours) (SpeedX8)
- # Actions: 12
- Computation Time: 0.001 secs
- Execution Time: 106 secs

MCTS (SpeedX8)
- # Actions: 12
- Computation Time: 0.047 secs
- Execution Time: 111 secs

BiRRT (fmRS) (SpeedX8)
- # Actions: 31
- Computation Time: 0.034 secs
- Execution Time: 267 secs
Comparative Study: Cuboids

TRLB(ours) (SpeedX8)

- # Actions: 11
- Computation Time: 0.011 secs
- Execution Time: 117 secs

BiRRT(fmRS) (SpeedX8)

- # Actions: 38
- Computation Time: 5.15 secs
- Execution Time: 394 secs
TRLB Performance on Challenging Cases

20-Cylinder Instance
# Actions: 22
Computation Time: 0.0023 secs

20-Cuboid Instance
# Actions: 26
Computation Time: 0.15 secs

ICRA 2022
# Actions: 15
Computation Time: 0.33 secs

Rutgers Robotics
# Actions: 17
Computation Time: 0.022 secs
Thank you for listening!
Fast High-Quality Tabletop Rearrangement in Bounded Workspace

Kai Gao\textsuperscript{1}  Darren Lau\textsuperscript{2}  Baichuan Huang\textsuperscript{1}
Kostas E. Bekris\textsuperscript{1}  Jingjin Yu\textsuperscript{1}

\textsuperscript{1}Computer Science \textdegree{} Rutgers University
\textsuperscript{2}Computer Science \textdegree{} Cornell University

Submitted to 2022 International Conference on Robotics and Automation (ICRA 2022)