Matching Outcomes in TTC vs. The Crawler

Conghui Bi

Nanjing audit university School of Economics, Nanjing 211815, Jiangsu, China

DOI: 10.32629/memf.v5i5.2854

Abstract: This paper compares the matching outcomes and mechanism characteristics of two algorithms: TTC (Gale's Top Trading Cycles) and the Crawler, under single-peaked preferences. The TTC algorithm matches through a cycle-pointing process, which theoretically guarantees efficiency and strategy-proofness, but in practice, it is more complex, particularly when implementing obviously dominant strategies. In contrast, the Crawler algorithm provides a more intuitive and easier-to-understand matching process by screening agents in order of house sizes, making it particularly suitable for environments with single-peaked preferences. This paper analyzes the strengths and weaknesses of both algorithms and explores their applicability and differences in matching outcomes across various scenarios. The conclusion drawn is that in the domain of single-peaked preferences, the Crawler algorithm offers significant advantages in terms of comprehension and implementation, while TTC, despite its broader applicability, poses higher operational complexity under specific preference structures.

Keywords: TTC; the Crawler; matching

1. Introduction

Matching theory is an important research area in economics and computer science, widely applied in real-world issues such as housing allocation, school admissions, and organ donation. The core goal of matching algorithms is to ensure that participants are assigned to their most preferred options while maintaining overall efficiency and fairness in the distribution.

Gale's Top Trading Cycles (TTC) algorithm, as a classic matching algorithm, achieves efficient matching between agents and objects through a cycle-pointing process. While TTC theoretically guarantees efficiency, strategy-proofness, and individual rationality, its implementation is complex, especially when it involves obviously dominant strategies. In contrast, the Crawler algorithm screens agents in order of house sizes, providing a more intuitive matching process, particularly showing significant advantages in single-peaked preference scenarios.

This study focuses on comparing the matching outcomes of TTC and Crawler algorithms under single-peaked preferences, analyzing their respective strengths and weaknesses, and exploring their applicability and performance differences in various application scenarios. Through a comparative study of these two algorithms, we aim to reveal their practical operational effects in single-peaked preference matching problems, providing reference for research and applications in related fields.

The paper will first introduce the basic principles and operational processes of the TTC and Crawler algorithms in detail, followed by a comparative analysis of their matching efficiency and strategy-proofness in a single-peaked preference environment. Finally, through specific case studies, the paper will demonstrate the performance of these two algorithms in practical applications. The research results will contribute to a deeper understanding of the applicability of matching algorithms under different preference structures.

2. Definitions

There is a set of agents $N := \{1, \ldots, n\}$. Each agent $i \in N$ is initially endowed with house i, so the set of houses is also N. Each agent i has a transitive and complete preference $\succsim_i$ over all houses. A profile of all agents' preferences is denoted $\succsim$A house j is $\succsim_i$-acceptable if $j \succsim_i i$. The preference $\succsim_i^{[j]}$ is picky about j if no house other than i and j is $\succsim_i^{[j]}$ acceptable. If $j \succsim_i j'$ holds for some $j \in N$ and all j' in some set $N' \subset N$, I write $j \succsim_i N'$. The names of all houses reflect an objective linear order on all houses. Houses are ordered by their sizes and $i < j$ means that i is smaller than j. The preference $\succsim_i$ is single-peaked (with respect to the order < on all houses) if there exists a $i^* \in N$ such that $j \succsim_i k$ holds if either $k < j \leq i^*$ or $k > j \geq i^*$. A preference is a linear order if it is antisymmetric. Arbitrary domains of agent i 's preferences and of preference profiles are denoted Ω_i and $\Omega := \Omega_1 \times \ldots, \times \Omega_n$. The domains of all linear preferences, of all singlepeaked preferences and of all linear, single-peaked preferences respectively are$\left\|\Omega^1, \widetilde{\widehat{\Omega}},\right\|$ and $\left\|\widehat{\Omega}^1 = \Omega^1 \cap \widehat{\Omega}\right.$. The initial endowment is represented by the matching $id : N \to N$ where $id(i) = i$ for all $i \in N$. Agents only care about their own houses, so agent i prefers matching μ to matching μ' if and only if $\mu(i) \succsim_i \mu'(i)$. A matching μ is efficient at $\succsim$ if $\mu(i)$ is $\succsim_i$-acceptable for each i. A social choice function $scf : \Omega \to M$ maps each profile $\succsim$ in the arbitrary domain Ω to a matching in M. Any social choice function can be viewed as a direct revelation mechanism with the understanding that each agent i declares a preference $\succsim_i$ to the designer, who in turn chooses the matching $scf(\succsim)$ given that $\succsim$ is the profile of stated preferences. The mechanism scf is efficient (individually rational) if $scf(\succsim)$ is efficient (individually rational) at $\succsim$ for each $\succsim \in \Omega$. It is strategy proof if no agent has an incentive to misrepresent his preferences, so $scf(\succsim)(i) \succsim_i scf(\succsim_i', \succsim -i)(i)$ holds for all $i, \succsim_i' \| and \succsim$.

2.1 Top trading cycles algorithm

Step l: Each agent points to the owner of his favorite house.

Due to the finiteness of agents, there exists at least one cycle (including self-cycles). More-over, cycles do not intersect.

Each agent in a cycle is assigned the house of the agent he points to and removed from the market.

If there is at least one remaining agent, proceed with the next step.

Step $\boldsymbol{k}$: Each remaining agent points to the owner of his favorite house among the remaining houses.

Each agent in a cycle is assigned the house of the agent he points to and removed from the market.

If there is at least one remaining agent, proceed with the next step.

End: No agents remain. It is clear that the algorithm will terminate within finite steps. Let Step t denote the last step.

Example 1: Top Trading Cycle

Consider a housing market with a set of agents $I = \{1,2,3,4\}$. Their preference profiles are as follows:

P_1	P_2	P_3	P_4
h_4	h_4	h_4	h_1
h_3	h_3	h_3	h_2
h_2	h_2	h_2	h_3
h_1	h_1	h_1	h_4

In the TTC mechanism , all agents select their most preferred houses from the set $\{h_1, h_2, h_3, h_4\}$. Thus, agents 1, 2, and 3 point to h_4 ,while agent 4 points to h_1. Consequently, agent 1 and 4 trade in a cycle, receiving h_4 and h_1 respectively. Subsequently, agents 3 and 2 match with h_3 and h_2 in successive steps. The final matching is $(4h_1, 2h_2, 3h_3, 1h_4)$.

2.2 The Crawler algorithm

Step 1: Each agent selects her most preferred house from those remaining. Among agents who point to their current house or to houses on their right, the rightmost agent pointing to a house is matched with that house and removed along with the house. Occupants of houses between the vacated and removed houses move to the next house on the left, treating it as their new endowment.

Step $\boldsymbol{k}, \boldsymbol{k} \geq 2$: This step is repeated for the remaining agents until all have been successfully matched.

Example 2: The Crawler algorithm

Consider a housing market with a set of agents $I = \{1,2,3,4\}$. Their preference profiles are as follows:

P_1	P_2	P_3	P_4
h_4	h_4	h_4	h_1
h_3	h_3	h_3	h_2
h_2	h_2	h_2	h_3
h_1	h_1	h_1	h_4

In the Crawler mechanism, no agent initially occupies their most preferred house. Agents 1,2, and 3 aim to move rightwards, with agent 3 being the rightmost. Hence, agent 3 is selected first, matched with h_4, causing agent 4 to"crawl" to h_3. In the next step, agent 2 is matched with h_3, and agent 4 "crawls" to h_2. Finally, agent 1 is matched with h_2, and agent 4 secures h_1. The Crawler's final matching is $(4h_1, 1h_2, 2h_3, 3h_4)$,which is equivalent to a sequential trading process ending with agent 4 receiving h_1.

3. Conclusion

In this study, we explored the performance of TTC and Crawler algorithms in addressing matching problems under single-peaked preferences. Through comparison, we found that the two algorithms exhibit significant differences in their matching sequences and priority-handling mechanisms, leading to distinct matching outcomes. Specifically, the TTC algorithm relies on cycles formed between agents, where each agent points to their most preferred house, facilitating exchanges among agents and ensuring that each agent receives their top choice. This mechanism emphasizes overall efficiency, but its complexity lies in the necessity of forming cycles, which can be challenging when implementing obviously dominant strategies.

In contrast, the Crawler algorithm employs a step-by-step screening and "crawling" approach. It screens agents in order of house sizes, prioritizing the rightmost agent for matching. After an agent is matched to a house, the remaining agents "crawl" sequentially to the next available house. This operational process makes the matching more intuitive, particularly in scenarios where agents gradually enter or exit the market. The Crawler algorithm demonstrates greater flexibility and efficiency in these dynamic environments.

This difference highlights the unique strengths and limitations of TTC and Crawler when addressing different preference structures and application scenarios. TTC is suitable for situations where all agents' preferences are known in advance and where maximizing overall efficiency is the primary goal. On the other hand, Crawler, with its flexible matching mechanism, is better suited for dynamic market environments, especially under single-peaked preferences. Therefore, when choosing an algorithm, one must consider the specific application context and preference structure to balance the strengths and weaknesses of both, ultimately achieving the most optimal matching outcome.

References

[1] Shapley, Lloyd, and Herbert Scarf. (1974), On cores and indivisibility. Journal of mathematical economics. 1, 23-37.
[2] Roth, A. (1982), Incentive compatibility in a market with indivisible goods, Economics letters, 9, 127-132.
[3] Ma, J. (1994), Strategyproofness and the Strict Core in a Market with Indivisibilites, International Journal of Game Theory, 23, 75-83.
[4] Bade, S. (2019), Matching with single-peaked preferences, Journal of Economic Theory, 180, 81-99.
[5] Li, S. (2017), Obviously Strategy-Proof Mechanisms, American Economic Review, 107, 3257-87.