capstract:2611.00006v1 [cs.AI] 4 Nov 2026

Planning for the Termination of Reciprocal Obligations

Elvira S. Montalbán*
Department of Intelligent Systems
Alborán Institute of Computation
e.montalban@alboran.example

Naveen K. Velanthur
Laboratory for Sequential Decision Systems
Cormandel School of Engineering

Tereza P. Jandourková
Department of Intelligent Systems
Alborán Institute of Computation
Centre for Formal Agency
Vltava Technical Institute

Abstract

A delivery action may satisfy an outstanding obligation and create a second obligation in the opposite direction. Planners that project a task onto possession alone can therefore report success in a state with an executable return action. We study this discrepancy in the return of borrowed serving dishes. We give an obligation ledger with separately observable custody and release, and characterize a class of return rules under which every compliant sequence preserves a positive outstanding balance. We then model authorization for empty return as an observation-dependent action and solve for a finite-horizon policy over the resulting belief states. In 800 held-out initial states, a ledger-aware policy completes 692 (86.5%), compared with 620 (77.5%) for maximum-posterior replanning, a paired difference of 9.0 percentage points (95% family-bootstrap interval 5.5 to 12.6). Under compulsory refill, closure is impossible even when scheduled exchanges always succeed. A benchmark extension in which one serving dish arrives inside another exposes cases where obtaining permission to return the inner dish supplies the occasion required to renew the outer obligation.

1 Introduction

A recipient borrows a serving dish when accepting food brought in it. Washing the dish and returning it satisfies the owner’s claim to the object. In the conventions considered here, the recipient also owes a gesture of return, usually food presented in the same dish. When that presentation counts as a gift, the original owner receives the dish and incurs the corresponding obligation. The dish is already home. The plan has another step.

We study a small family of return rules extracted as executable predicates from descriptions of reciprocal service (Orellana Sieder and Hult, 2019; Dameri and Audenot, 2021). The unit under study is the return convention. The rules are inputs, and a violation is a disagreement with an input rule. For the simplest convention, we prove that the number of outstanding gestures stays constant over all permitted handoffs. We use that condition to remove search branches that have lost access to release. For the stochastic part of the model, an authorization action exposes information about the other party’s willingness to close. The policy has to reserve a subsequent visit for the physical return. A late authorization can arrive after the last visit it could have authorized.

*Corresponding author: e.montalban@alboran.example

Preprint. Under review.

2 Related work

2.1 Representation and search

The distinction between an action precondition and a persistent fluent follows STRIPS (Fikes and Nilsson, 1971). A return can have one delete effect and one add effect on the same kind of fluent. In our domain the arguments of those effects are reversed, so merging them as a net zero edit to custody discards information. For temporal control we use availability windows over discrete events. Allen's interval relations (Allen, 1983) can express finer scheduling constraints. We use only precedence and co-occurrence, which also lets us compile most conventions into the state-dependent operators described by Ghallab et al. (2004). The clock-dependent release rule is explicit in that compilation. Temporal search control (Bacchus and Kabanza, 2000) would be a natural way to specify prohibitions involving an entire sequence of solicitations.

Classical heuristic search over a projection is inexpensive (Bonet and Geffner, 2001), and delete-relaxation heuristics (Hoffmann and Nebel, 2001) are particularly good at finding the next meeting that both parties can attend. The relaxation admits plans that keep the dish at two addresses at once and plans that still own a signed release after it has expired. Strong cyclic planning allows a fair nondeterministic loop with a possible exit (Cimatti et al., 2003). Here the outcome of the next inquiry depends on an unchanging disposition and a changing calendar. Treating a standing refusal as a repeatable draw from a fair outcome set can certify a policy that will leave the same dish outstanding for every remaining step. Žuvela and Vind (2020) found a related dependence when evaluating repeated requests for confirmation of whether a loan was permanent.

2.2 Planning under incomplete information

Our observation kernel makes a subproblem a mixed-observability POMDP (Kaelbling et al., 1998). The finite-horizon objective follows the backward induction of Smallwood and Sondik (1973). We use a loss for leaving debt outstanding at the deadline and a per-step cost. An undiscounted shortest-path formulation would require care about improper policies (Bertsekas and Tsitsiklis, 1991). Under compulsory refill there is no proper policy from a nonempty ledger. In domestic restoration tasks, earlier simulators assign an end-of-episode reward when every object is at its recorded home (Avero and Diko, 2021; Leitão and Bregger, 2023). Reversal of a debt has no effect on that reward. Sereke and Nolet (2022) distinguish acknowledgment of an object from acknowledgment of an accompanying contribution. Their physical-return benchmarks record both, though they end each episode after the first admitted transfer.

3 Domain and observability

3.1 Ledger

Let $o(d)$ be the owner of dish d and $p(d)$ its current possessor. The two roles are properties of the object and the execution, respectively. A directed token $u = (d, i, j)$ says that i owes j a gesture on account of d. A token never serves as authority to change the owner field. At initialization the gift has already been accepted, the receiving party holds the object, and one token exists per active dish. An unfilled transfer requires authorization for the return to the owner. Tokens stay in the ledger until consumed by a handoff. A signed permission changes what is legal at a meeting, with no immediate change to the ledger.

In the one-dish fragment a filled handoff consumes (d, i, j) and adds (d, j, i). The new debtor also takes custody. For the purposes of this fragment an item is filled if the receiver counts its contents as a gift, even if no new food was prepared. Reclassifying a gift after the exchange is outside the domain. A permitted unfilled handoff consumes the token and leaves $p(d) = j$. If j is a borrower the operator also places the object in the owner's custody, within the same meeting, when the owner is present and the contract permits direct return. The planner can solicit permission for that three-party handoff. For a single pair, a release that would deliver an object away from its owner is unavailable. For a two-person return this restriction blocks release after an odd number of filled handoffs.

Let $\mathcal{D}$ be the set of dishes and U_t the outstanding tokens after transition t. The end test given to the evaluator is

$$G_t = \mathbf{1}\{U_t = \varnothing\} \prod_{d \in \mathcal{D}} \mathbf{1}\{p_t(d) = o(d)\}. \tag{1}$$

3.2 Invariant under refill

Let $r(a)$ count tokens consumed by executed action a and $g(a)$ count tokens created. We count creation events even when the new token has the same arguments as a previously resolved token. By construction,

$$|U_{t+1}| - |U_t| = g(a_t) - r(a_t). \tag{2}$$

Proposition 1 (Closed refill rules)**.** *Suppose $|U_0| > 0$ and each available action that consumes a token creates at least as many tokens. Then no finite execution achieves Eq. (1).*

Proof. For transitions that consume a token, the hypothesis gives $g(a) \geq r(a)$. For all others $r(a) = 0$ and $g(a) \geq 0$. Summing Eq. (2) gives $|U_t| \geq |U_0|$ at every finite t. Apply Eq. (1). ∎

When meetings remain available, one filled handoff restores custody in the two-person, one-dish case. Another restores the initial ledger and replaces the dish at the borrower's address. On a repeating calendar even the augmented state repeats if the next available meeting is indexed modulo the calendar period. In the lifted representation the two most recent gifts have different event identifiers. Those identifiers may be ignored for reachability. They may not be ignored for costing, since the policy pays for each gift regardless of which direction it went.

3.3 Requests and release

The domain has two request channels. A private inquiry produces either an indication of willingness or an indication of refusal. It grants no action permission. A formal request can produce a signed ticket for the next valid visit or a refusal. The ticket identifies the dish, its final owner, its current possessor and the visit slot. A signed ticket satisfies the release precondition only for the named exchange. The planner can leave it unused. It cannot be spent early on a different visit or used twice. The rule compiler rejects tickets for visits with incompatible custody requirements, even when the named people are all free.

Each pair has a latent disposition z sampled once, with three classes. An unconditional class accepts formal requests for a feasible return slot. A calendar-dependent class accepts them when the occasion meets the active rule's predicate. A continuing class declines. These are deterministic conditions on the formal action and latent state. The private inquiry is noisy conditional on z and the occasion family. At the default prototype it returns a positive sign with probabilities 0.85, 0.60 and 0.15, respectively. Rule templates draw perturbations of at most 0.05 per row. A group extending the private reply without extending the validity of its tickets changes two rows of the observation kernel and no row of the transition kernel. We ablate the private channel in Section 6.3.

Time is public. A wait action advances to the next calendar slot, and every other action advances by one slot after it is processed. Speaking slots and transfer slots can overlap. Because each step holds one action, asking for a release during a transfer slot forfeits the transfer at that slot. Each feasible successor is computed after expiring all permission records dated earlier than the new time. Expiration is independent of whether the planner has a later return scheduled.

4 Planning

Write s for the public state, including the ledger and the remaining calendar, and $b(z)$ for the posterior over dispositions. The formal response is observable. Given response y to action a, the update is

$$b^{a,y}(z) = \frac{b(z) P(y \mid s, a, z)}{\sum_{z'} b(z') P(y \mid s, a, z')}. \tag{3}$$

We use the exact posterior at the current public state as the key, with the remaining horizon, in a memoized dynamic program. All probabilities are rational and are stored as reduced integers during planning. We choose a cost of 1 per executed action, an additional cost of 2 per filled transfer, and a terminal cost of 32 for an open case. The first two costs describe event counts, with a premium for producing a gift. A demand to minimize gifts alone admits a policy that waits until time runs out. Under the stated costs a policy is allowed to stop early and take the penalty if the reachable closure probability is too low. Illegal actions are masked in the ledger-aware variants.

If $c(a)$ is one-step cost and $s^{a,y}$ the public successor, the full search uses

$$V_h(s,b) = \min\left\{32,\ \min_a \left[c(a) + \sum_y P(y \mid s,a,b) V_{h-1}(s^{a,y}, b^{a,y})\right]\right\}. \tag{4}$$

Terminal goals have value zero for every horizon, and $V_0(s,b) = 32$ for other states. Actions restricted by a valid ticket are generated before Eq. (3) is called. The sum includes all observable outcomes of actions with nonzero probability. In Algorithm 1 the certificate check first ignores all private replies and sets each formal reply optimistically to acceptance wherever some element of posterior support permits it. It searches this finite relaxed graph for a goal within the remaining time. For the closed refill fragment Proposition 1 proves the same result without building the graph. The relaxed test can underestimate the cost of successful closure. It cannot remove a state from which a legal closure is reachable within the horizon.

Algorithm 1 Finite-horizon search with an optimistic closure test

Require: public state s, disposition posterior b, remaining slots h
if $G(s) = 1$ **then**
  return 0
end if
if $h = 0$ **or** RelaxedClosure$(s, \mathrm{supp}(b), h)$ fails **then**
  return 32
end if
$v \leftarrow 32$
for each admissible action a at s **do**
  $q \leftarrow c(a)$
  for each response y with positive predictive probability **do**
    $q \leftarrow q + P(y \mid s,a,b) V_{h-1}(s^{a,y}, b^{a,y})$
  end for
  $v \leftarrow \min(v, q)$
end for
return v

5 Evaluation design

5.1 Instance construction

We generated 600 independent rule families. A family fixes a rule graph, availability mask, observation-noise range and ownership pattern. Four start cases per family vary the initial slot and draw latent dispositions from the family's prior. Cases in a family share their rule graph and public time grid. We assigned 300 families to development, 100 to validation and 200 to the test set before generating the four cases. The test set has 50 families of each of four kinds. A rule family's salted identifier fixes its split, and a separate stream keyed by the instance identifier drives observation outcomes. We fixed tie breaks, cost weights and the 12-slot horizon before the test runs. Appendix A describes the split and the random streams. Test-time priors are provided to every belief-based policy. Outcome draws for a given action, state and visit are coupled across policies; alternate visits receive their own draws. The random seed never enters a policy state.

The four case kinds appear in Table 1. In the ordinary case, a pair with one dish has periodic meetings that can support release. In the calendar case, the release is tied to one marked visit in a shorter window. The pair may receive two independently owed dishes in the nested case. The outer dish can contain the inner dish during transfer, and custody records refer to each dish separately. A formal request for the inner return may name a visit that satisfies the calendar predicate for a pending outer gift. That visit also turns on the corresponding refill operator, even for a party that could previously perform only an empty transfer at that address. In the inherited case some admissible signatures bind a previous owner’s availability pattern and some bind the active owner. Present ownership is always public. Inheritance is a stored rule on a particular dish, not something inferred from a family name.

Table 1: Test starts and valid closures under ledger-aware search. Each row contains 50 distinct rule families.

Rule family	Dishes	Starts	Closures	Rate (%)
Ordinary	1	200	190	95.0
Calendar	1	200	183	91.5
Nested	2	200	172	86.0
Inherited	1	200	147	73.5
All		800	692	86.5

5.2 Comparators and inference

Inventory A* uses a cost of one per step with an admissible lower bound given by the number of needed deliveries divided by the maximum dishes delivered at a visit. It commits to an empty return because the added gift premium makes a filled handoff strictly more expensive for the same delivered custody. Refill only excludes all release actions and executes the earliest permitted filled return when custody is wrong. If custody has been restored it stops. Myopic release scores each request by its predicted acceptance and the earliest compatible delivery. If no request can secure a ticket for an upcoming visit, it falls back to the next feasible filled return that restores custody. MAP replanning solves the deterministic problem for the posterior mode at each step and masks returns for which it lacks an actual signed ticket. The disposition oracle uses the same cost and exact search after receiving z. It still needs a signed ticket. Informing the oracle about existing permissions alone would add no information, since all policies can read their own records.

The primary estimand is the test-family mean paired difference in valid closure probability between full search and MAP replanning. A valid closure occurs when all actions pass replay and the first $G_t = 1$ occurs by slot 12. Early stops with open debt, and planned actions rejected by replay, count as failures. We report percentile intervals from 10,000 bootstrap replicates of families within each case kind (Efron and Tibshirani, 1993). Within each replicate we keep the two policies paired on each start. Resampling individuals would split cases that share their availability masks. Comparisons beyond the primary contrast are descriptive. The terminal-cost ablation reuses the same event streams.

6 Results

6.1 Completion and possession

The inventory-only plan returns all objects in 782 cases in its projected simulator, which accepts physically feasible handoffs without reading the ledger. In the full domain those 782 plans attempt an unlicensed transfer, and the other 18 never find a delivery slot. The refill policy has lower physical recovery and zero closure. Table 2 gives counts, and Figure 1 displays the gap on the proportion scale. The three release policies and the oracle have identical admissibility masks. Their differences reflect which visits and outcomes they make reachable. The oracle has the highest observed completion and should be read as a comparator supplied with privileged information, not a pointwise upper bound on success. It minimizes expected cost under Eq. (4), which allows early stopping.

Table 2: Outcomes by planner at horizon 12. Closure requires a legal trace. Inventory custody is measured in the custody-only simulator.

Planner	Custody restored	Valid closures	Closure (%)	Open or invalid
Inventory A*	782	0	0.0	800
Refill only	761	0	0.0	800
Myopic release	671	600	75.0	200
MAP replanning	693	620	77.5	180
Full search	734	692	86.5	108
Disposition oracle	755	724	90.5	76

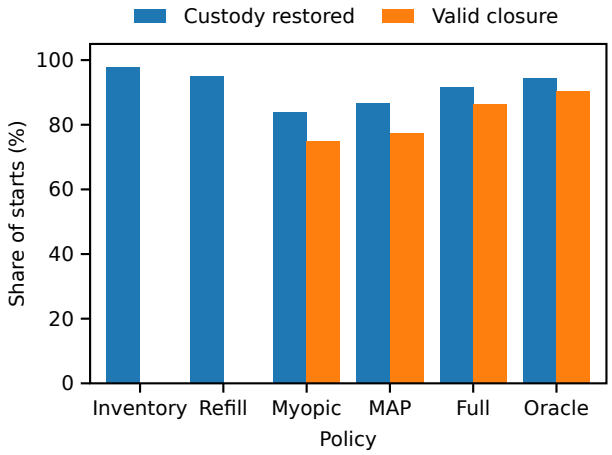


Figure 1: Physical restoration and valid closure by policy. The inventory policy's restoration count uses its projected simulator.

The paired difference against MAP replanning is 9.0 points (95% interval 5.5 to 12.6). Full search finds a productive solicitation in some cases where the modal type is continuing. The same solicitation can be worth the slot when the posterior puts its remaining mass on a calendar-dependent type and the only qualifying visit is two slots away. Conversely, for a pair whose unconditional class is modal only until the first private inquiry, a plan that spends the speaking slot on a formal request can leave too little time to act on the refusal it obtains. The main difference in the logs is the placement of the first formal request. In a paired audit of the 72 net additional closures, there were 103 cases closed by full search alone and 31 closed by MAP alone. Several of the latter had a substantially lower posterior closure probability than typical cases at initialization. The cost criterion permits declining such a case even when its sampled disposition would have allowed closure.

6.2 When the projected plan ends

Figure 2 gives cumulative completion under the fixed 12-slot policy, rather than results from solving a different instance at each horizon. At slot four, 286 valid closures have occurred. The inventory policy has restored 708 projected inventories by that slot. A success metric triggered by the last reduction in physical distance would therefore put the inventory planner well ahead early in the episode. The gap peaks at 422 cases at slot four on this grid.

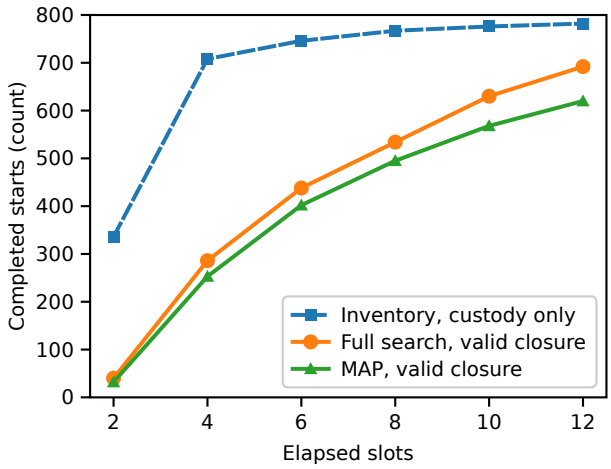


Figure 2: Cumulative closures in the fixed twelve-slot runs. Inventory A* is credited in the upper series when all objects reach their owners in its projection.

We traced the 108 failures of full search to the first event that made closure impossible or to a voluntary halt while closure still had positive probability. The resulting categories are mutually exclusive. A refusal that revealed the continuing type accounted for 47, expiration of the last feasible visit for 39, and a positive-cost voluntary halt for 22. The first two categories use the latent-state replay when assigning cause. In the third category the halt precedes the deadline, and an alternative allowed policy would have a positive probability of closing from the public state. An oracle cannot necessarily change the first category. It can change some calendar expirations by asking for the appropriate recipient’s signature before the other party has responded.

Figure 3 shows the classification by case kind. The inherited failure rate includes both continuing-type cases and brittle calendars that no longer agree with current possession. In 16 of the 53 inherited failures the last possible slot was an occasion attached to the past owner. Six of those slots had already elapsed before the dish was actually returned to the active owner with a gift. At that point the new obligation was attached to a valid owner and an appointment that could only occur in the past.

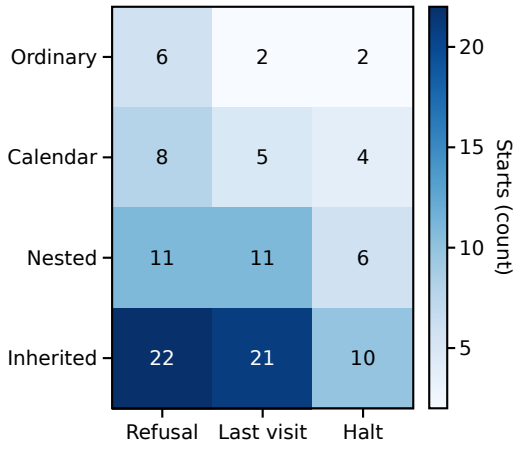


Figure 3: First classified failure per open execution under full search. Rows sum to the failures implied by the denominators in the case table.

6.3 Policy ablations

Removing the private channel costs 26 closures. Raising the terminal cost to 64 recovers ten at the expense of increasing the mean executed actions per start from 6.1 to 6.7. The search pursues favorable replies with low prior probability when the cost of stopping rises. A third ablation lets a ticket name two successive visit slots while still requiring all relevant parties to accept it. Only the nested family has a second physically feasible visit. Table 3 includes this result as a change to the legal execution set.

Table 3: Policy and rule sensitivity at the same initial states. The extended ticket changes the valid execution set.

Setting	Closures	Difference
Full search	692	0
No private channel	666	−26
Terminal cost 64	702	+10
Extended nested ticket	705	+13

7 Nested transfers and accumulated precedent

A plate can travel on a tray. The physical movement of the plate then requires custody of both items, even if the tray remains an independent loan. Benarroch and Kvasný (2024) reports that borrowers sometimes present the small container already sitting inside the larger one when the recipient agreed to receive only the small container. We introduce a variant with an explicit support constraint. An inner dish may be handed over by itself when it has been removed beforehand. Removal occupies a slot. A gift on the tray counts against the tray’s own token. An outer token is therefore resolvable with an unfilled handoff only if the tray reaches its owner without a departure gift, including a filled inner dish resting on it.

We instrumented 60 separately generated nested instances with a side rule for acknowledgments (Fraser and Caras, 2022). The recipient at the designated release visit can sign a receipt and send it back in a small envelope. A receipt in the recipient’s hand creates nothing. A receipt presented on the outer tray counts as a contribution under the active outer rule. We varied that routing bit while holding the signature, container state and visit constant. A valid policy closed 49 of 60 starts when the receipt was handed over separately and 32 of 60 when it traveled on the tray. On replay, 18 starts closed only under separate routing and one only under tray routing. The exceptional case with tray routing uses a new outer token to schedule a visit the original release did not make available. The class is small enough that we report counts without using it in the primary inference.

The episode that motivated this extension began with a cake plate placed on a borrowed tray. The designated return visit accepted the empty tray. The acknowledgment for receiving it was handed back on the now-empty tray. From the viewpoint of the acknowledgment rule the recipient had just received a signed document as a presented object and owed a gesture. The owner lent out the tray again to carry the acknowledgment home. By Eq. (2) the two-document variant cannot terminate if each acknowledgment subsequently travels on an object whose rule treats it as a contribution. The physical size of the acknowledgment is absent from the proof. Printing it smaller helps only on instances whose support rule admits a different carrier.

8 Scope and conclusion

The generator takes return conventions as fixed for an episode. A recipient can decline a release, but cannot stop counting deliveries halfway through unless the template admits that transition. The fractions above quantify solver performance under those inputs. No prevalence estimate for any disposition follows from them. Established relations may also share a release across multiple objects. The two-dish model with one ticket per dish has no such global release. Adding the operator would require a joint authorization state when the people who own the dishes differ. Matsuzaki and

Bile (2023) describes that case for organized lending, where a local custodian signs for a collection under a shared rule.

A Generator and implementation

Each rule family has one of four topology codes, its sequence of 12 availability masks and a latent-type prior tabulated in twentieths. The generator draws at least two feasible visit masks per actor before applying the template restrictions. A latent draw uses integers from a counter-based stream; observation thresholds use hundredths and the same stream with a separately keyed counter. Calendar masks for development and validation can have the same Hamming weight as test masks. They cannot have the same generator template identifier. Split assignment is stratified by topology. We canonicalize actor labels before generating test ids to keep relabelings of a single case together. Multiple starts per family are retained when their realized dispositions or starting times coincide. There is no case-wise deduplication after the split.

B Search cost and reproducibility

We checked the relaxed certificate by disabling it on the validation set and comparing the selected action and value at every start. Search tables are process-local and discarded between starts. At the main horizon the median end-to-end full-search time per case was 8.4 ms, and the 95th percentile was 62 ms, on one 3.2 GHz x86-64 CPU core. These timings include the reachability precheck and exclude replay. Search evaluates up to five action schemas per slot in the core generator, with grounded arguments generating additional branches in two-dish cases. Peak resident memory for a single sequential test sweep was 74 MB. The inner receipt extension runs independently with the same state representation and a routing bit in the operator input. No learned model or accelerator is used.

Acknowledgments

We thank Flóra Eszenyi for independently implementing the transfer replay and Márcio D. Calvetti for identifying the support constraint in the two-dish generator.

References

James F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, 1983.

Cristóbal M. Avero and Mellanie Diko. Home Positions as Terminal Predicates in Domestic Task Planning. In *Workshop on Computational Models of Domestic Activity*, pages 51–59, 2021.

Fahiem Bacchus and Froduald Kabanza. Using Temporal Logics to Express Search Control Knowledge for Planning. *Artificial Intelligence*, 116(1–2):123–191, 2000.

Marina E. Benarroch and Samuel Kvasný. Stacked Containers and the Scope of Return Permission. *Review of Material Exchange*, 17:177–201, 2024.

Dimitri P. Bertsekas and John N. Tsitsiklis. An Analysis of Stochastic Shortest Path Problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.

Blai Bonet and Hector Geffner. Planning as Heuristic Search. *Artificial Intelligence*, 129(1–2):5–33, 2001.

Alessandro Cimatti, Marco Pistore, Marco Roveri, and Paolo Traverso. Weak, Strong, and Strong Cyclic Planning via Symbolic Model Checking. *Artificial Intelligence*, 147(1–2):35–84, 2003.

Giuliano T. Dameri and Aline Audenot. Discharge Conditions in Reciprocal Service. *Studies in Economic Practice*, 19(2):114–139, 2021.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, 1993.

Richard E. Fikes and Nils J. Nilsson. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence*, 2(3–4):189–208, 1971.

Moira G. Fraser and Doina V. Caras. Document Transfer as a Credited Contribution. *Journal of Transactional Practice*, 11(3):248–269, 2022.

Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann, 2004.

Jörg Hoffmann and Bernhard Nebel. The FF Planning System: Fast Plan Generation Through Heuristic Search. *Journal of Artificial Intelligence Research*, 14:253–302, 2001.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and Acting in Partially Observable Stochastic Domains. *Artificial Intelligence*, 101(1–2):99–134, 1998.

Marta V. Leitão and Ole H. Bregger. Receipt and Restoration in Episodic Object Exchange. *Journal of Situated Planning*, 8:206–231, 2023.

Rento Matsuzaki and Mariette F. Bile. Delegated Authority in Collective Equipment Lending. *Studies in Economic Practice*, 21:53–77, 2023.

Beatriz Orellana Sieder and Mirja R. Hult. Vessels in Circulation: Property Retention in Small-Scale Food Exchange. *Review of Material Exchange*, 12:21–48, 2019.

Luwam Sereke and Céline R. Nolet. Returns with Compound Acknowledgment. In *Workshop on Representing Social Action*, pages 84–92, 2022.

Richard D. Smallwood and Edward J. Sondik. The Optimal Control of Partially Observable Markov Processes over a Finite Horizon. *Operations Research*, 21(5):1071–1088, 1973.

Daria M. Žuvela and Asger R. Vind. Deferred Decisions in Household Lending Protocols. In *Workshop on Temporal Agency*, pages 38–46, 2020.