Human review packet: LOS02CombinatorialAuctions

Generated from byte-pinned source excerpts, the expanded PaperInterface specifications, and hash-bound raw-source-to-Spec screening records.

Paper: LOS02CombinatorialAuctions
Paper id: LOS02CombinatorialAuctions
Source version: Journal of the ACM 49(5), 2002
Source record: audit/paper_statement_map.json
Rows: 12
Generated: 2026-09-06
Source URL: official source

How to use this packet

For each source claim, compare the exact source excerpts with the expanded PaperInterface specification. Mark up this PDF or fill the blank reviewer box in the TeX source; return the annotated file for reconciliation into the paper-local human-review log.

Open the interactive dashboard

The interactive dashboard is an optional alternative to using this PDF; it presents the same review surface rather than an additional review requirement. After forking and cloning (or pulling) AppliedModelingLib, run the following from the repository root. The cache command is needed only the first time in a new clone.

```
cd <your-repository-checkout>
lake exe cache get
python3 scripts/review_dashboard.py --paper LOS02CombinatorialAuctions --serve
```

Open http://127.0.0.1:8765/ in a browser and keep that terminal running while reviewing. The dashboard records saved annotations in this paper's local reviewer-owned review record.

Contents

- Semantic library prerequisites (5; not paper claims)
  - AppliedModelingLib.Auction.Bundle
  - AppliedModelingLib.Auction.CombinatorialAuction.utility
  - AppliedModelingLib.Auction.SingleMindedBid.bundleSize
  - AppliedModelingLib.Auction.allocationValue
  - AppliedModelingLib.FairDivision.Bundle
- Paper-specific semantic prerequisites (18; not paper claims)
  - LOS02CombinatorialAuctions.NegativeResults.GreenTruthfulPayment
  - LOS02CombinatorialAuctions.NegativeResults.GreenType.ReportAdmissible
  - LOS02CombinatorialAuctions.NegativeResults.GreenType.TrueAdmissible
  - LOS02CombinatorialAuctions.NegativeResults.requestedBundle
  - LOS02CombinatorialAuctions.NegativeResults.section8Witness
  - LOS02CombinatorialAuctions.SourceCritical.Critical
  - LOS02CombinatorialAuctions.SourceCritical.Feasible
  - LOS02CombinatorialAuctions.SourceCritical.LegalProfile
  - LOS02CombinatorialAuctions.SourceCritical.Monotonicity
  - LOS02CombinatorialAuctions.SourceCritical.Participation
  - LOS02CombinatorialAuctions.SourceCritical.SqrtNormNoTies
  - LOS02CombinatorialAuctions.SourceCritical.TotalValue
  - LOS02CombinatorialAuctions.SourceCritical.Valuation
  - LOS02CombinatorialAuctions.SourceGVA.Admissible
  - LOS02CombinatorialAuctions.SourceGVA.MaximizesFeasible
  - LOS02CombinatorialAuctions.SourceGVA.Truthful
  - LOS02CombinatorialAuctions.SourceGreedy.averageGreedyMechanism
  - LOS02CombinatorialAuctions.SourceGreedy.averagePayment
- Source claims (12)
  - theorem4_1Spec

- proposition4_2Spec
- theorem7_2Spec
- section8Spec
- lemma9_1Spec
- lemma9_2Spec
- lemma9_3Spec
- lemma9_4Spec
- lemma9_5Spec
- theorem9_6Spec
- theorem10_2Spec
- section12Spec

Material library prerequisites

AppliedModelingLib.Auction.Bundle

Source locator: cited publication, lines 205-225

Verbatim paper-source connection

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will
range over the bidders. Bidders are selfish, but rational, and trying to maximize their
utility in the final outcome. A bidder knows his own utility function (i.e., his type),
but this information is private and neither the auctioneer nor the other players have
access to it. The final result of an auction consists of two elements: an allocation
of the goods and a vector of payments from the bidders to the auctioneer, both of
which are functions of the bidders' declarations (i.e., bids). Formally, we have a
finite set G of k goods and an allocation is a partial function from G to P, that is, a
function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all
goods be allocated. Notice that the allocations produced by the Generalized Vickrey
Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always
total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions
from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types
for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice
that such a type allows for both complementarity and substitutability, but not for
externalities. Since the set 2i does not depend on i, we shall write 2. An element
of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is
also the set of messages that bidder i may send. A bidder may send any element of
2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t
to denote a (true) type, d to denote a message, T or D to denote vectors of n types
and P for a payment vector, that is, a vector of n nonnegative numbers.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will
range over the bidders. Bidders are selfish, but rational, and trying to maximize their
utility in the final outcome. A bidder knows his own utility function (i.e., his type),
but this information is private and neither the auctioneer nor the other players have
access to it. The final result of an auction consists of two elements: an allocation
of the goods and a vector of payments from the bidders to the auctioneer, both of
which are functions of the bidders' declarations (i.e., bids). Formally, we have a
finite set G of k goods and an allocation is a partial function from G to P, that is, a
function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all
goods be allocated. Notice that the allocations produced by the Generalized Vickrey
Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always
total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions
from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types
for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice
that such a type allows for both complementarity and substitutability, but not for
externalities. Since the set 2i does not depend on i, we shall write 2. An element
of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is
also the set of messages that bidder i may send. A bidder may send any element of
2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t
to denote a (true) type, d to denote a message, T or D to denote vectors of n types
and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly
standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G
and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

Let us denote the bundle obtained by i as:
gi (D) = f (D)−1 (i).

(2)

Lean-expanded library semantic target

type:
Type u_1 → Type u_1

value:
fun (Item : Type u_1) => AppliedModelingLib.FairDivision.Bundle Item

Exact Lean library declaration

/-- Bundles in combinatorial auctions reuse the fair-division bundle model. -/
abbrev Bundle (Item : Type*) := FairDivision.Bundle Item

Recorded source-to-library screening **Verdict:** matches

Reason: The pinned allocation-domain text fixes a finite good set G and uses bundles s ⊆ G. The expanded target is Finset Item (through FairDivision.Bundle), which, with Item instantiated by G, is exactly a finite collection of distinct goods representing such a bundle. It asserts no extra allocation or feasibility property.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

AppliedModelingLib.Auction.CombinatorialAuction.utility

Source locator: cited publication, lines 226-231

Verbatim paper-source connection

Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

(1)

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

Definition 5.1. Bidder i is single-minded if and only if there is a set s ⊂ G of goods and a value v ∈ R+ such that its type t can be described as: t(s 0) = v if s ⊆ s 0 and t(s 0) = 0 otherwise.

[Next verbatim source excerpt]

The general structure of the properties of interest is that they consider a given set of single-minded types and vary one of those types. They restrict the changes that can appear in the allocation or the payments as a result of such a change. Let declarations be fixed, but arbitrary, for all bidders except j. Consider two possible declarations for j: hs, vi and hs 0 , v 0 i. Given an allocation scheme f and a payment scheme p, we shall consider the allocations and payments generated by both declarations of j. Let gi be the set of goods obtained by bidder i if j declares hs, vi, and gi0 the set he obtains if j declares hs 0 , v 0 i. Similarly denote by pi and pi0 the payments of i.

Lean-expanded library semantic target

```
type:
{Bidder : Type u_1} →
  {Item : Type u_2} →
    AppliedModelingLib.Auction.CombinatorialAuction Bidder Item →
      AppliedModelingLib.Auction.CombinatorialReport Bidder Item →
        AppliedModelingLib.Auction.CombinatorialReport Bidder Item → Bidder → ℝ

value:
fun {Bidder : Type u_1} {Item : Type u_2} (M : AppliedModelingLib.Auction.CombinatorialAuction Bidder Item)
    (values reports : AppliedModelingLib.Auction.CombinatorialReport Bidder Item) (i : Bidder) =>
  values i (M.allocation reports i) - M.payment reports i
```

Exact Lean library declaration

```
/-- Quasilinear utility for a combinatorial-auction bidder. -/
def utility (M : CombinatorialAuction Bidder Item)
    (values reports : CombinatorialReport Bidder Item) (i : Bidder) : ℝ :=
  values i (M.allocation reports i) - M.payment reports i
```

Recorded source-to-library screening **Verdict:** matches

Reason: Equation (1) in the pinned quasilinear-utility text is u = t(s) − x. The expanded target evaluates bidder i's supplied value function at the bundle allocated from reports and subtracts that bidder's payment: values i (M.allocation reports i) − M.payment reports i. Thus values, the allocated bundle, and payment occupy the same roles as t, s, and x.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

AppliedModelingLib.Auction.SingleMindedBid.bundleSize

Source locator: cited publication, lines 630-652

Verbatim paper-source connection

a
.
Definition 7.1. The average amount per good of a bid b = hs, ai is |s|

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will
range over the bidders. Bidders are selfish, but rational, and trying to maximize their
utility in the final outcome. A bidder knows his own utility function (i.e., his type),
but this information is private and neither the auctioneer nor the other players have
access to it. The final result of an auction consists of two elements: an allocation
of the goods and a vector of payments from the bidders to the auctioneer, both of
which are functions of the bidders' declarations (i.e., bids). Formally, we have a
finite set G of k goods and an allocation is a partial function from G to P, that is, a
function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all
goods be allocated. Notice that the allocations produced by the Generalized Vickrey
Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always
total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions
from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types
for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice
that such a type allows for both complementarity and substitutability, but not for
externalities. Since the set 2i does not depend on i, we shall write 2. An element
of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is
also the set of messages that bidder i may send. A bidder may send any element of
2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t
to denote a (true) type, d to denote a message, T or D to denote vectors of n types
and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly
standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G
and payment x is:
u = t(s) − x.

Lean-expanded library semantic target

type:
{Item : Type u_2} → AppliedModelingLib.Auction.SingleMindedBid Item → ℝ

value:
fun {Item : Type u_2} (b : AppliedModelingLib.Auction.SingleMindedBid Item) => ↑(Finset.card b.desired)

Exact Lean library declaration

```
/-- The size of a single-minded desired bundle, as a real number. -/
def bundleSize (b : SingleMindedBid Item) : ℝ :=
  (b.desired.card : ℝ)
```

Recorded source-to-library screening **Verdict:** matches

Reason: Pinned Definition 7.1 writes a bid as ⟨s,a⟩ and uses |s| for the number of goods in its requested bundle. The expanded target is the real coercion of Finset.card b.desired, precisely that bundle-cardinality quantity. It is the cardinality prerequisite, not a claim that it alone is the full a / |s| ratio.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

AppliedModelingLib.Auction.allocationValue

Source locator: cited publication, lines 282-341

Verbatim paper-source connection

In a GVA, the allocation chosen maximizes the sum of the declared valuations of the bidders, each bidder receives a monetary amount that equals the sum of the declared valuations of all other bidders, and pays the auctioneer the sum of such valuations that would have been obtained if he had not participated in the auction. A way to describe such an auction, in which i does not participate, is to consider the auction in which bidder i declares a zero valuation for all possible bundles. A bidder with zero valuation for all bundles has no influence on the outcome. Formally, given a vector D of declarations, the generalized Vickrey auction defines the allocation and payment policies as follows (notice that a −1 (i) is the bundle allocated to i by allocation a, and that gi is defined in Eq. (2)):
f (D) = argmaxa∈O

n
X
i=1

di (a −1 (i)),

(4)

[form-feed in source extraction]
Approximately Efficient Combinatorial Auctions
n
X

p j (D) = −

n
X

di (gi (D)) +

i=1,i6= j

583

di (gi (Z)),

(5)

i=1,i6= j

where Z i = Di for any i 6= j and Z j (s) = 0 for any bundle s ⊆ G. Since d j (g j (Z)) = 0, we may as well have written:
p j (D) = −

n
X

di (gi (D)) +

i=1,i6= j

n
X

di (gi (Z)).

(6)

i=1

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

Lean-expanded library semantic target

```
type:
{Bidder : Type u_1} →
  {Item : Type u_2} →
    [Fintype Bidder] →
      AppliedModelingLib.Auction.CombinatorialReport Bidder Item →
        AppliedModelingLib.Auction.BundleAllocation Bidder Item → ℝ

value:
fun {Bidder : Type u_1} {Item : Type u_2} [Fintype Bidder]
    (values : AppliedModelingLib.Auction.CombinatorialReport Bidder Item)
    (A : AppliedModelingLib.Auction.BundleAllocation Bidder Item) =>
  ∑ i : Bidder, values i (A i)
```

Exact Lean library declaration

```
/-- Social welfare of an allocation under reported bundle values. -/
noncomputable def allocationValue [Fintype Bidder]
    (values : CombinatorialReport Bidder Item)
    (A : BundleAllocation Bidder Item) : ℝ :=
  ∑ i : Bidder, values i (A i)
```

Recorded source-to-library screening **Verdict:** matches

Reason: The pinned GVA definition in equation (4) maximizes the sum over bidders of declared valuations $d_i(a^{-1}(i))$. The expanded target maps a declared-value profile and allocation to $\sum$ i, values i (A i), the same declared-value sum evaluated at each bidder's allocated bundle. The Fintype binder supplies the source's finite bidder index set.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

AppliedModelingLib.FairDivision.Bundle

Source locator: cited publication, lines 205-225

Verbatim paper-source connection

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

Let us denote the bundle obtained by i as:
gi (D) = f (D)−1 (i).

(2)

Lean-expanded library semantic target

type:
Type u_1 → Type u_1

value:
fun (Item : Type u_1) => Finset Item

Exact Lean library declaration

/-- A bundle of indivisible goods. -/
abbrev Bundle (Item : Type*) := Finset Item

Recorded source-to-library screening **Verdict:** matches

Reason: The pinned allocation-domain text uses a bundle as a subset s of the finite good set G. The expanded target is Finset Item, a finite duplicate-free collection of items; at Item = G it represents exactly such a bundle. Its FairDivision namespace does not alter that underlying source concept.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Paper-specific semantic prerequisites

LOS02CombinatorialAuctions.NegativeResults.GreenTruthfulPayment

Paper-local declaration:
LOS02CombinatorialAuctions.NegativeResults.GreenTruthfulPayment
Source locator: cited publication, lines 1152-1166

Verbatim paper-source connection

Assume there are two goods a and b and two bidders Green and Red. Red is
single-minded and his type is h{a}, 10i. Red truthfully declares his type. Green is
interested in both b and the set {a, b}. His valuation is 0 for any allocation in which

[form-feed in source extraction]
598

D. LEHMANN ET AL.

he does not get b. It is v b for any allocation in which he gets b but not a, and it
is v ab if he gets both a and b, v ab > v b . Green's declaration is 0 for all allocations
that do not give him b, gb for all allocations that give him b but not a and gab for
the allocation in which he gets both a and b. Notice that four parameters describe
the auction: gab , gb , v ab and v b . Assume, furthermore, that 0 ≤ gb < 10. We reason
by contradiction and assume there is a payment scheme that makes truth-telling a
dominant strategy for Green. Let us consider two cases.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will
range over the bidders. Bidders are selfish, but rational, and trying to maximize their
utility in the final outcome. A bidder knows his own utility function (i.e., his type),
but this information is private and neither the auctioneer nor the other players have
access to it. The final result of an auction consists of two elements: an allocation
of the goods and a vector of payments from the bidders to the auctioneer, both of
which are functions of the bidders' declarations (i.e., bids). Formally, we have a
finite set G of k goods and an allocation is a partial function from G to P, that is, a
function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all
goods be allocated. Notice that the allocations produced by the Generalized Vickrey
Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always
total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions
from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types
for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice
that such a type allows for both complementarity and substitutability, but not for
externalities. Since the set 2i does not depend on i, we shall write 2. An element
of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is
also the set of messages that bidder i may send. A bidder may send any element of
2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t
to denote a (true) type, d to denote a message, T or D to denote vectors of n types
and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly
standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G
and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

could consider any super-set of the single-minded types that grows only exponentially with k. A very natural idea is to consider players that send off
↪ single-minded
agent bidders to do their work. The agents play rationally, but individually, and
bring the goods and the payments due to the player. In the final analysis, a player
gets all the goods obtained by each of his agents and pays all the payments imposed
on each of his agents. A player's strategy is then a small (i.e., polynomial in k) set
of single-minded agents (i.e., bids).

[Next verbatim source excerpt]

Definition 3.2. A mechanism h f, pi is truthful if and only if for every i ∈ P,
t ∈ 2 and any vector D of declarations, if D 0 is the vector obtained from D by
replacing the ith coordinate di by t, then: t(gi (D 0)) − pi (D 0) ≥ t(gi (D)) − pi (D).

[Next verbatim source excerpt]

A single-minded bidder declaring hs, ai, with s ⊆ G and a ∈ R+ will be said to
put out a bid b = hs, ai. We shall use s(b) and a(b) to denote the components of b
and call a(b) the amount of the bid b. As explained in Section 5, we identify bids
and bidders. Two bids b = hs, ai and b0 = hs 0 , a 0 i conflict if s ∩ s 0 6= ∅.
The algorithms we consider execute in two phases.
—In the first phase, the bids are sorted by some criterion. The algorithms of the
family are distinguished by the different criteria they use. Since there are n bids,
this phase takes time of the order of n log n. We assume a criterion, that is, a norm
is defined and the bids are sorted in decreasing order following this norm. Since
we shall have, in Theorem 10.2, to compare the sorted lists of bids of slightly
different auctions, we also assume a consistent treatment of ties, that is, bids
with equal norms. Formally, we shall assume that no two different bids have the
same norm, that is, there are no ties.
—In the second phase, a greedy algorithm generates an allocation. Let L be the
list of sorted bids obtained in the first phase. The first bid of L, say b = hs, ai is
granted, that is, the set s will be allocated to b and then the algorithm examines
each bid of L, in order, and grants it if it does not conflict with any of the bids
previously granted. If it does, it denies (i.e., does not grant) the bid. This phase
requires time linear in n.

[Next verbatim source excerpt]

a
.
Definition 7.1. The average amount per good of a bid b = hs, ai is |s|

Lean-elaborated paper signature

type:
(LOS02CombinatorialAuctions.NegativeResults.GreenType → ℝ) → Prop

value:
```
fun (payment : LOS02CombinatorialAuctions.NegativeResults.GreenType → ℝ) =>
  ∀ (values : LOS02CombinatorialAuctions.NegativeResults.GreenType),
    values.TrueAdmissible →
      ∀ (report : LOS02CombinatorialAuctions.NegativeResults.GreenType),
        report.ReportAdmissible →
          values.valuation (LOS02CombinatorialAuctions.NegativeResults.greenAllocation report) - payment report ≤
            values.valuation (LOS02CombinatorialAuctions.NegativeResults.greenAllocation values) - payment values
```

Recorded source-to-declaration screening

Verdict: matches
Reason: Section 12 fixes Red and compares Green's truthful utility with every permitted Green declaration; together with Section 3's quasilinear utility, the target exactly makes a report-only payment and requires no permitted report to improve on the true-value report. The separate true-value and report predicates retain the source's distinct v and g roles.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

LOS02CombinatorialAuctions.NegativeResults.GreenType.ReportAdmissible

Paper-local declaration:
LOS02CombinatorialAuctions.NegativeResults.GreenType.ReportAdmissible
Source locator: cited publication, lines 1152-1166

Verbatim paper-source connection

Assume there are two goods a and b and two bidders Green and Red. Red is single-minded and his type is h{a}, 10i. Red truthfully declares his type. Green is interested in both b and the set {a, b}. His valuation is 0 for any allocation in which

[form-feed in source extraction]
598

D. LEHMANN ET AL.

he does not get b. It is v b for any allocation in which he gets b but not a, and it is v ab if he gets both a and b, v ab > v b . Green's declaration is 0 for all allocations that do not give him b, gb for all allocations that give him b but not a and gab for the allocation in which he gets both a and b. Notice that four parameters describe the auction: gab , gb , v ab and v b . Assume, furthermore, that 0 ≤ gb < 10. We reason by contradiction and assume there is a payment scheme that makes truth-telling a dominant strategy for Green. Let us consider two cases.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

could consider any super-set of the single-minded types that grows only exponentially with k. A very natural idea is to consider players that send off
↪ single-minded
agent bidders to do their work. The agents play rationally, but individually, and bring the goods and the payments due to the player. In the final analysis, a player gets all the goods obtained by each of his agents and pays all the payments imposed on each of his agents. A player's strategy is then a small (i.e., polynomial in k) set of single-minded agents (i.e., bids).

[Next verbatim source excerpt]

Definition 3.2. A mechanism h f, pi is truthful if and only if for every i ∈ P, t ∈ 2 and any vector D of declarations, if D 0 is the vector obtained from D by replacing the ith coordinate di by t, then: t(gi (D 0)) − pi (D 0) ≥ t(gi (D)) − pi (D).

[Next verbatim source excerpt]

A single-minded bidder declaring hs, ai, with s ⊆ G and a ∈ R+ will be said to put out a bid b = hs, ai. We shall use s(b) and a(b) to denote the components of b and call a(b) the amount of the bid b. As explained in Section 5, we identify bids and bidders. Two bids b = hs, ai and b0 = hs 0 , a 0 i conflict if s ∩ s 0 6= ∅.
The algorithms we consider execute in two phases.
—In the first phase, the bids are sorted by some criterion. The algorithms of the family are distinguished by the different criteria they use. Since there are n bids, this phase takes time of the order of n log n. We assume a criterion, that is, a norm is defined and the bids are sorted in decreasing order following this norm. Since we shall have, in Theorem 10.2, to compare the sorted lists of bids of slightly different auctions, we also assume a consistent treatment of ties, that is, bids with equal norms. Formally, we shall assume that no two different bids have the same norm, that is, there are no ties.
—In the second phase, a greedy algorithm generates an allocation. Let L be the list of sorted bids obtained in the first phase. The first bid of L, say b = hs, ai is granted, that is, the set s will be allocated to b and then the algorithm examines each bid of L, in order, and grants it if it does not conflict with any of the bids previously granted. If it does, it denies (i.e., does not grant) the bid. This phase requires time linear in n.

[Next verbatim source excerpt]

a
.
Definition 7.1. The average amount per good of a bid b = hs, ai is |s|

Lean-elaborated paper signature

type:
LOS02CombinatorialAuctions.NegativeResults.GreenType → Prop

value:
fun (v : LOS02CombinatorialAuctions.NegativeResults.GreenType) => 0 ≤ v.both ∧ 0 ≤ v.bOnly ∧ v.bOnly < 10

Recorded source-to-declaration screening **Verdict:** matches

Reason: Section 12 states 0 ≤ g_b < 10 for Green's declaration, while the Section 3 message domain makes every declared bundle value nonnegative. Thus nonnegative g_ab and g_b with g_b < 10 are exactly the selected declaration slice.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

LOS02CombinatorialAuctions.NegativeResults.GreenType.TrueAdmissible

Paper-local declaration:
LOS02CombinatorialAuctions.NegativeResults.GreenType.TrueAdmissible
Source locator: cited publication, lines 1152-1166

Verbatim paper-source connection

Assume there are two goods a and b and two bidders Green and Red. Red is single-minded and his type is h{a}, 10i. Red truthfully declares his type. Green is interested in both b and the set {a, b}. His valuation is 0 for any allocation in which

[form-feed in source extraction]
598

D. LEHMANN ET AL.

he does not get b. It is v b for any allocation in which he gets b but not a, and it is v ab if he gets both a and b, v ab > v b . Green's declaration is 0 for all allocations that do not give him b, gb for all allocations that give him b but not a and gab for the allocation in which he gets both a and b. Notice that four parameters describe the auction: gab , gb , v ab and v b . Assume, furthermore, that 0 ≤ gb < 10. We reason by contradiction and assume there is a payment scheme that makes truth-telling a dominant strategy for Green. Let us consider two cases.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

could consider any super-set of the single-minded types that grows only exponentially with k. A very natural idea is to consider players that send off
↪ single-minded
agent bidders to do their work. The agents play rationally, but individually, and bring the goods and the payments due to the player. In the final analysis, a player gets all the goods obtained by each of his agents and pays all the payments imposed on each of his agents. A player's strategy is then a small (i.e., polynomial in k) set of single-minded agents (i.e., bids).

[Next verbatim source excerpt]

Definition 3.2. A mechanism h f, pi is truthful if and only if for every i ∈ P, t ∈ 2 and any vector D of declarations, if D 0 is the vector obtained from D by replacing the ith coordinate di by t, then: t(gi (D 0)) − pi (D 0) ≥ t(gi (D)) − pi (D).

[Next verbatim source excerpt]

A single-minded bidder declaring hs, ai, with s ⊆ G and a ∈ R+ will be said to put out a bid b = hs, ai. We shall use s(b) and a(b) to denote the components of b and call a(b) the amount of the bid b. As explained in Section 5, we identify bids and bidders. Two bids b = hs, ai and b0 = hs 0 , a 0 i conflict if s ∩ s 0 6= ∅.
The algorithms we consider execute in two phases.
—In the first phase, the bids are sorted by some criterion. The algorithms of the family are distinguished by the different criteria they use. Since there are n bids, this phase takes time of the order of n log n. We assume a criterion, that is, a norm is defined and the bids are sorted in decreasing order following this norm. Since we shall have, in Theorem 10.2, to compare the sorted lists of bids of slightly different auctions, we also assume a consistent treatment of ties, that is, bids with equal norms. Formally, we shall assume that no two different bids have the same norm, that is, there are no ties.
—In the second phase, a greedy algorithm generates an allocation. Let L be the list of sorted bids obtained in the first phase. The first bid of L, say b = hs, ai is granted, that is, the set s will be allocated to b and then the algorithm examines each bid of L, in order, and grants it if it does not conflict with any of the bids previously granted. If it does, it denies (i.e., does not grant) the bid. This phase requires time linear in n.

[Next verbatim source excerpt]

a
.
Definition 7.1. The average amount per good of a bid b = hs, ai is |s|

Lean-elaborated paper signature

type:
LOS02CombinatorialAuctions.NegativeResults.GreenType → Prop

value:
fun (v : LOS02CombinatorialAuctions.NegativeResults.GreenType) => 0 ≤ v.bOnly ∧ v.bOnly < v.both

Recorded source-to-declaration screening **Verdict:** matches

Reason: The Section 12 model gives Green nonnegative values and states v_ab > v_b. The target's 0 ≤ bOnly and bOnly < both expresses those conditions; nonnegativity of both follows from them.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

LOS02CombinatorialAuctions.NegativeResults.requestedBundle

Paper-local declaration:
LOS02CombinatorialAuctions.NegativeResults.requestedBundle
Source locator: cited publication, lines 1152-1166

Verbatim paper-source connection

Assume there are two goods a and b and two bidders Green and Red. Red is single-minded and his type is h{a}, 10i. Red truthfully declares his type. Green is interested in both b and the set {a, b}. His valuation is 0 for any allocation in which

[form-feed in source extraction]
598

D. LEHMANN ET AL.

he does not get b. It is v b for any allocation in which he gets b but not a, and it is v ab if he gets both a and b, v ab > v b . Green's declaration is 0 for all allocations that do not give him b, gb for all allocations that give him b but not a and gab for the allocation in which he gets both a and b. Notice that four parameters describe the auction: gab , gb , v ab and v b . Assume, furthermore, that 0 ≤ gb < 10. We reason by contradiction and assume there is a payment scheme that makes truth-telling a dominant strategy for Green. Let us consider two cases.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

could consider any super-set of the single-minded types that grows only exponentially with k. A very natural idea is to consider players that send off
↪ single-minded
agent bidders to do their work. The agents play rationally, but individually, and bring the goods and the payments due to the player. In the final analysis, a player gets all the goods obtained by each of his agents and pays all the payments imposed on each of his agents. A player's strategy is then a small (i.e., polynomial in k) set of single-minded agents (i.e., bids).

[Next verbatim source excerpt]

Definition 3.2. A mechanism h f, pi is truthful if and only if for every i ∈ P, t ∈ 2 and any vector D of declarations, if D 0 is the vector obtained from D by replacing the ith coordinate di by t, then: t(gi (D 0)) − pi (D 0) ≥ t(gi (D)) − pi (D).

[Next verbatim source excerpt]

A single-minded bidder declaring hs, ai, with s ⊆ G and a ∈ R+ will be said to put out a bid b = hs, ai. We shall use s(b) and a(b) to denote the components of b and call a(b) the amount of the bid b. As explained in Section 5, we identify bids and bidders. Two bids b = hs, ai and b0 = hs 0 , a 0 i conflict if s ∩ s 0 6= ∅.
The algorithms we consider execute in two phases.
—In the first phase, the bids are sorted by some criterion. The algorithms of the family are distinguished by the different criteria they use. Since there are n bids, this phase takes time of the order of n log n. We assume a criterion, that is, a norm is defined and the bids are sorted in decreasing order following this norm. Since we shall have, in Theorem 10.2, to compare the sorted lists of bids of slightly different auctions, we also assume a consistent treatment of ties, that is, bids with equal norms. Formally, we shall assume that no two different bids have the same norm, that is, there are no ties.
—In the second phase, a greedy algorithm generates an allocation. Let L be the list of sorted bids obtained in the first phase. The first bid of L, say b = hs, ai is granted, that is, the set s will be allocated to b and then the algorithm examines each bid of L, in order, and grants it if it does not conflict with any of the bids previously granted. If it does, it denies (i.e., does not grant) the bid. This phase requires time linear in n.

[Next verbatim source excerpt]

a
.
Definition 7.1. The average amount per good of a bid b = hs, ai is |s|

Lean-elaborated paper signature

type:
Fin 3 → Finset (Fin 2)

value:
fun (a : Fin 3) => ![{0}, {0, 1}, {1}] a

Recorded source-to-declaration screening

Verdict: matches
Reason: The Section 12/Example 8.1 source instance has requested bundles a, {a,b}, and b in that order for the three bid positions. The Fin 2/Fin 3 target maps exactly to {0}, {0,1}, and {1}.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

LOS02CombinatorialAuctions.NegativeResults.section8Witness

Paper-local declaration:
LOS02CombinatorialAuctions.NegativeResults.section8Witness
Source locator: cited publication, lines 759-769

Verbatim paper-source connection

In Section 10, a mechanism based on the greedy allocation will be built and shown to be truthful if all bidders are single-minded. In this section, we show that the use of Clarke's payment scheme, used in the GVA and described in Eq. (5), in conjunction with the greedy allocation does not make for a truthful mechanism, even if bidders

[form-feed in source extraction]
590

D. LEHMANN ET AL.

are single-minded. In other terms, if the greedy allocation and Clarke's payment scheme are used, a bidder may have an incentive to lie about his valuation. The

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

A single-minded bidder declaring hs, ai, with s ⊆ G and a ∈ R+ will be said to put out a bid b = hs, ai. We shall use s(b) and a(b) to denote the components of b and call a(b) the amount of the bid b. As explained in Section 5, we identify bids and bidders. Two bids b = hs, ai and b0 = hs 0 , a 0 i conflict if s ∩ s 0 6= ∅.
The algorithms we consider execute in two phases.
—In the first phase, the bids are sorted by some criterion. The algorithms of the family are distinguished by the different criteria they use. Since there are n bids, this phase takes time of the order of n log n. We assume a criterion, that is, a norm is defined and the bids are sorted in decreasing order following this norm. Since we shall have, in Theorem 10.2, to compare the sorted lists of bids of slightly different auctions, we also assume a consistent treatment of ties, that is, bids with equal norms. Formally, we shall assume that no two different bids have the same norm, that is, there are no ties.
—In the second phase, a greedy algorithm generates an allocation. Let L be the list of sorted bids obtained in the first phase. The first bid of L, say b = hs, ai is granted, that is, the set s will be allocated to b and then the algorithm examines each bid of L, in order, and grants it if it does not conflict with any of the bids previously granted. If it does, it denies (i.e., does not grant) the bid. This phase requires time linear in n.

[Next verbatim source excerpt]

a
.
Definition 7.1. The average amount per good of a bid b = hs, ai is |s|

[Next verbatim source excerpt]

In a GVA, the allocation chosen maximizes the sum of the declared valuations of the bidders, each bidder receives a monetary amount that equals the sum of the declared valuations of all other bidders, and pays the auctioneer the sum of such valuations that would have been obtained if he had not participated in the auction. A way to describe such an auction, in which i does not participate, is to consider the auction in which bidder i declares a zero valuation for all possible bundles. A bidder with zero valuation for all bundles has no influence on the outcome. Formally, given a vector D of declarations, the generalized Vickrey auction defines the allocation and payment policies as follows (notice that a −1 (i) is the bundle allocated to i by allocation a, and that gi is defined in Eq. (2)):
f (D) = argmaxa∈O

n
X

i=1

di (a −1 (i)),

(4)

[form-feed in source extraction]

n
X

p j (D) = −

n
X

di (gi (D)) +

i=1,i6= j

583

di (gi (Z)),

(5)

i=1,i6= j

where Z i = Di for any i 6= j and Z j (s) = 0 for any bundle s ⊆ G. Since
d j (g j (Z)) = 0, we may as well have written:
p j (D) = −

n
X

di (gi (D)) +

i=1,i6= j

n
X

di (gi (Z)).

(6)

i=1

[Next verbatim source excerpt]

Definition 5.1. Bidder i is single-minded if and only if there is a set s ⊂ G of goods and a value v ∈ R+ such that its type t can be described as: t(s 0) = v if s ⊆ s 0 and t(s 0) = 0 otherwise.

[Next verbatim source excerpt]

Example 8.1. As in Example 7.3, there are two goods a and b and three bidders Red, Green, and Blue. Red bids 10 for a, Green bids 19 for the set {a, b} and Blue bids 8 for b. The greedy algorithm grants Red's and Blue's bids and denies Green's bid, that is, f (D)(a) = Red and f (D)(b) = Blue. We shall now compute Red's payment. For this allocation, we have the following declared valuations: v Blue = 8 and v Green = 0. If Red had bid zero, the greedy algorithm would have granted Green's bid and denied Blue's bid. Therefore, the allocation f (Z) is defined by: f (Z)(a) = f (Z)(b) = Green, where v Blue = 0 and v Green = 19. Clarke's payment scheme gives to Red: 8 − 0 for Blue and 0 − 19 for Green, that is, Red pays 11.

Lean-elaborated paper signature

type:
Prop

value:
```
LOS02CombinatorialAuctions.NegativeResults.greedyClarke.utility
      (LOS02CombinatorialAuctions.NegativeResults.threeBids 10 19 8)
      (LOS02CombinatorialAuctions.NegativeResults.threeBids 10 19 8) 0 =
    -1 ∧
  LOS02CombinatorialAuctions.NegativeResults.greedyClarke.utility
      (LOS02CombinatorialAuctions.NegativeResults.threeBids 10 19 8)
      (LOS02CombinatorialAuctions.NegativeResults.threeBids 0 19 8) 0 =
    0
```

Recorded source-to-declaration screening

Verdict: matches
Reason: Example 8.1 reports that Red's truthful greedy-plus-Clarke run has payment 11 and utility 10−11=−1, whereas the zero report rerun has utility zero. The target is precisely those two runs, with the original type retained as the utility valuation in both.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

LOS02CombinatorialAuctions.SourceCritical.Critical

Paper-local declaration:
LOS02CombinatorialAuctions.SourceCritical.Critical
Source locator: cited publication, lines 888-897

Verbatim paper-source connection

Our third property deals with a satisfied bidder: a satisfied bidder pays exactly the critical value of Lemma 9.1, that is, the lowest value he could have declared and still be allocated the goods he desires.
Critical

s ⊆ g j ⇒ p j = vc

Notice that Critical says, first, that the payment for a bid that is granted does not depend on the amount of the bid, it depends only on the other bids. Then it says that it is exactly equal to the critical value below which the bid would have lost.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

Definition 5.1. Bidder i is single-minded if and only if there is a set s ⊂ G of goods and a value v ∈ R+ such that its type t can be described as: t(s 0) = v if s ⊆ s 0 and t(s 0) = 0 otherwise.

[Next verbatim source excerpt]

The general structure of the properties of interest is that they consider a given set of single-minded types and vary one of those types. They restrict the changes that can appear in the allocation or the payments as a result of such a change. Let declarations be fixed, but arbitrary, for all bidders except j. Consider two possible declarations for j: hs, vi and hs 0 , v 0 i. Given an allocation scheme f and a payment scheme p, we shall consider the allocations and payments generated by both declarations of j. Let gi be the set of goods obtained by bidder i if j declares hs, vi, and gi0 the set he obtains if j declares hs 0 , v 0 i. Similarly denote by pi and pi0 the payments of i.

[Next verbatim source excerpt]

LEMMA 9.1. In a mechanism that satisfies Exactness and Monotonicity, given a bidder j, a set s of goods and declarations for all other bidders, there exists a critical value v c such that
∀v, v < v c ⇒ g j = ∅,
∀v, v > v c ⇒ g j = s,
We allow v c to be infinite if f (As,v)−1 (j) = ∅ for every v. Note that we do not know whether j's bid is granted or not in case v = v c .

[Next verbatim source excerpt]

Our next property, Monotonicity, also concerns only the allocation scheme. It requires that, if j's bid is granted if he declares hs, vi, it is also granted if he declares hs 0 , v 0 i for any s 0 ⊆ s, v 0 ≥ v. In other words, proposing more money for fewer goods cannot cause a bidder to lose his bid. It follows that, similarly, offering less money for more goods cannot cause a lost bid to win. Formally:
Monotonicity s ⊆ g j ,

s 0 ⊆ s,

v 0 ≥ v ⇒ s 0 ⊆ g 0j .

Lean-elaborated paper signature

type:
{Bidder : Type u} →

```
  {Item : Type v} → AppliedModelingLib.Auction.SingleMindedAcceptedMechanism Bidder Item → Type (max u v)

value:
fun {Bidder : Type u} {Item : Type v} (M : AppliedModelingLib.Auction.SingleMindedAcceptedMechanism Bidder Item) =>
  M.NonnegativeCriticalValueWithInfinityCertificate
```

Recorded source-to-declaration screening

Verdict: matches

Reason: Section 9 defines the critical value as the lowest declaration still winning, permits an infinite value when every value loses, and leaves equality at the threshold unresolved. The displayed certificate has finite strict-below denial/ strict-above acceptance, an infinite-denial alternative, own-bid independence, and accepted-payment equality, without adding an equality outcome.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

LOS02CombinatorialAuctions.SourceCritical.Feasible

Paper-local declaration:
LOS02CombinatorialAuctions.SourceCritical.Feasible
Source locator: cited publication, lines 840-844

Verbatim paper-source connection

Exactness Either g j = s or g j = ∅.
In an exact allocation, we shall say that j's bid is granted in the first case, and denied in the second case. In such a scheme, the allocation may be viewed as a set of bids (or bidders) that is conflict-free, that is, the s coordinates have pairwise empty intersections. A GVA, as we defined it, does not in fact always satisfy Exactness.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

Definition 5.1. Bidder i is single-minded if and only if there is a set s ⊂ G of goods and a value v ∈ R+ such that its type t can be described as: t(s 0) = v if s ⊆ s 0 and t(s 0) = 0 otherwise.

[Next verbatim source excerpt]

The general structure of the properties of interest is that they consider a given set of single-minded types and vary one of those types. They restrict the changes that can appear in the allocation or the payments as a result of such a change. Let declarations be fixed, but arbitrary, for all bidders except j. Consider two possible declarations for j: hs, vi and hs 0 , v 0 i. Given an allocation scheme f and a payment scheme p, we shall consider the allocations and payments generated by both declarations of j. Let gi be the set of goods obtained by bidder i if j declares hs, vi, and gi0 the set he obtains if j declares hs 0 , v 0 i. Similarly denote by pi and pi0 the payments of i.

Lean-elaborated paper signature

type:
{Bidder : Type u_1} → {Item : Type u_2} → AppliedModelingLib.Auction.SingleMindedAcceptedMechanism Bidder Item → Prop

value:
```
fun {Bidder : Type u_1} {Item : Type u_2} (M : AppliedModelingLib.Auction.SingleMindedAcceptedMechanism Bidder Item) =>
  ∀ (D : Bidder → AppliedModelingLib.Auction.SingleMindedBid Item),
    LOS02CombinatorialAuctions.SourceCritical.LegalProfile D →
      AppliedModelingLib.Auction.PairwiseDisjointDesired D (M.accepted D)
```

Recorded source-to-declaration screening **Verdict:** matches

Reason: Section 9 explains that an exact allocation can be viewed as a conflict-free set of bids. Because the supporting accepted-set mechanism allocates each accepted bidder its desired bundle and each other bidder nothing, pairwise disjoint desired bundles on legal profiles is exactly that source feasibility content.

Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

LOS02CombinatorialAuctions.SourceCritical.LegalProfile

Paper-local declaration:
LOS02CombinatorialAuctions.SourceCritical.LegalProfile
Source locator: cited publication, lines 453-455

Verbatim paper-source connection

Definition 5.1. Bidder i is single-minded if and only if there is a set s ⊂ G of
goods and a value v ∈ R+ such that its type t can be described as: t(s 0) = v if s ⊆ s 0
and t(s 0) = 0 otherwise.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will
range over the bidders. Bidders are selfish, but rational, and trying to maximize their
utility in the final outcome. A bidder knows his own utility function (i.e., his type),
but this information is private and neither the auctioneer nor the other players have
access to it. The final result of an auction consists of two elements: an allocation
of the goods and a vector of payments from the bidders to the auctioneer, both of
which are functions of the bidders' declarations (i.e., bids). Formally, we have a
finite set G of k goods and an allocation is a partial function from G to P, that is, a
function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all
goods be allocated. Notice that the allocations produced by the Generalized Vickrey
Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always
total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions
from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types
for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice
that such a type allows for both complementarity and substitutability, but not for
externalities. Since the set 2i does not depend on i, we shall write 2. An element
of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is
also the set of messages that bidder i may send. A bidder may send any element of
2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t
to denote a (true) type, d to denote a message, T or D to denote vectors of n types
and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly
standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G
and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

better approximations if agents are assumed to be single-minded and one parameter. We shall denote by hs, vi the type just described. Note that a
↪ single-minded
bidder enjoys free disposal. We shall assume, in most of this article, that all bidders
are single-minded, that is, there are sets of goods si and nonnegative real numbers
v i such that bidder i is of type hsi , v i i. We shall denote by 6 the set of all singleminded types. The size of the set 6, contrary to the size of 2, is singly
↪ exponential
in k. A string of polynomial size will be enough to code the declarations of the
bidders: it will describe a set of goods and a value. In this setting, we identify bids

[Next verbatim source excerpt]

a
.
Definition 7.1. The average amount per good of a bid b = hs, ai is |s|

[Next verbatim source excerpt]

LEMMA 9.2. In a mechanism that satisfies Exactness and Participation, a
bidder whose bid is denied has utility zero.

Lean-elaborated paper signature

type:
{Bidder : Type u_1} → {Item : Type u_2} → (Bidder → AppliedModelingLib.Auction.SingleMindedBid Item) → Prop

value:
fun {Bidder : Type u_1} {Item : Type u_2} (bids : Bidder → AppliedModelingLib.Auction.SingleMindedBid Item) =>
  AppliedModelingLib.Auction.SingleMindedAcceptedMechanism.NonnegativeNonemptyProfile bids

Recorded source-to-declaration screening **Verdict:** matches

Reason: Definition 5.1 supplies a single-minded desired set and a nonnegative value. In the selected Sections 7, 9, and 10 average-greedy scope, the target's nonempty desired-set condition makes the printed per-good norms defined while its remaining condition is exactly nonnegative single-minded bids; it introduces no source-level equality rule.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

LOS02CombinatorialAuctions.SourceCritical.Monotonicity

Paper-local declaration:
LOS02CombinatorialAuctions.SourceCritical.Monotonicity
Source locator: cited publication, lines 848-857

Verbatim paper-source connection

Our next property, Monotonicity, also concerns only the allocation scheme. It requires that, if j's bid is granted if he declares hs, vi, it is also granted if he declares hs 0 , v 0 i for any s 0 ⊆ s, v 0 ≥ v. In other words, proposing more money for fewer goods cannot cause a bidder to lose his bid. It follows that, similarly, offering less money for more goods cannot cause a lost bid to win. Formally:
Monotonicity s ⊆ g j ,

s 0 ⊆ s,

v 0 ≥ v ⇒ s 0 ⊆ g 0j .

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

Definition 5.1. Bidder i is single-minded if and only if there is a set s ⊂ G of goods and a value v ∈ R+ such that its type t can be described as: t(s 0) = v if s ⊆ s 0 and t(s 0) = 0 otherwise.

[Next verbatim source excerpt]

The general structure of the properties of interest is that they consider a given set of single-minded types and vary one of those types. They restrict the changes that can appear in the allocation or the payments as a result of such a change. Let declarations be fixed, but arbitrary, for all bidders except j. Consider two possible declarations for j: hs, vi and hs 0 , v 0 i. Given an allocation scheme f and a payment scheme p, we shall consider the allocations and payments generated by both declarations of j. Let gi be the set of goods obtained by bidder i if j declares hs, vi, and gi0 the set he obtains if j declares hs 0 , v 0 i. Similarly denote by pi and pi0 the payments of i.

[Next verbatim source excerpt]

LEMMA 9.4. In a mechanism that satisfies Exactness, Monotonicity, Participation and Critical, a bidder j of type hs, vi is never better off declaring hs, v 0 i for some v 0 6= v than by being truthful.

Lean-elaborated paper signature

```
type:
{Bidder : Type u_1} → {Item : Type u_2} → AppliedModelingLib.Auction.SingleMindedAcceptedMechanism Bidder Item → Prop

value:
fun {Bidder : Type u_1} {Item : Type u_2} (M : AppliedModelingLib.Auction.SingleMindedAcceptedMechanism Bidder Item) =>
  M.MonotonicityOn LOS02CombinatorialAuctions.SourceCritical.LegalProfile
```

Recorded source-to-declaration screening **Verdict:** matches

Reason: Section 9 says that a granted ⟨s,v⟩ remains granted after declaring ⟨s',v'⟩ for s' ⊆ s and v' ≥ v. MonotonicityOn LegalProfile has exactly those weak subset/value inequalities and requires both the original and changed declaration profiles to remain in the stated source domain.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

LOS02CombinatorialAuctions.SourceCritical.Participation

Paper-local declaration:
LOS02CombinatorialAuctions.SourceCritical.Participation
Source locator: cited publication, lines 906-913

Verbatim paper-source connection

Our last property concerns the payment scheme. Together with Critical, it implies that the utility of no truthful bidder is negative. It concerns unsatisfied bidders, bids that are denied. We require that an unsatisfied bidder pay zero. The utility of an unsatisfied bidder is then zero. This is simply tuning the utility scales of the different bidders, or, ensuring that bidders may not lose by participating in the auction.
Participation

s 6⊆ g j ⇒ p j = 0

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

Definition 5.1. Bidder i is single-minded if and only if there is a set s ⊂ G of goods and a value v ∈ R+ such that its type t can be described as: t(s 0) = v if s ⊆ s 0 and t(s 0) = 0 otherwise.

[Next verbatim source excerpt]

The general structure of the properties of interest is that they consider a given set of single-minded types and vary one of those types. They restrict the changes that can appear in the allocation or the payments as a result of such a change. Let declarations be fixed, but arbitrary, for all bidders except j. Consider two possible declarations for j: hs, vi and hs 0 , v 0 i. Given an allocation scheme f and a payment scheme p, we shall consider the allocations and payments generated by both declarations of j. Let gi be the set of goods obtained by bidder i if j declares hs, vi, and gi0 the set he obtains if j declares hs 0 , v 0 i. Similarly denote by pi and pi0 the payments of i.

Lean-elaborated paper signature

type:
{Bidder : Type u_1} → {Item : Type u_2} → AppliedModelingLib.Auction.SingleMindedAcceptedMechanism Bidder Item → Prop

value:
fun {Bidder : Type u_1} {Item : Type u_2} (M : AppliedModelingLib.Auction.SingleMindedAcceptedMechanism Bidder Item) =>
  ∀ (D : Bidder → AppliedModelingLib.Auction.SingleMindedBid Item),
    LOS02CombinatorialAuctions.SourceCritical.LegalProfile D → ∀ i ∉ M.accepted D, M.payment D i = 0

Recorded source-to-declaration screening **Verdict:** matches

Reason: Section 9's Participation condition is that an unsatisfied bid pays zero. Under the supporting exact accepted-set allocation and nonempty legal requests, unsatisfied is equivalent to not being accepted, so the target's zero-payment clause is the same property on the source declaration domain.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

LOS02CombinatorialAuctions.SourceCritical.SqrtNormNoTies

Paper-local declaration:
LOS02CombinatorialAuctions.SourceCritical.SqrtNormNoTies
Source locator: cited publication, lines 591-617

Verbatim paper-source connection

—In the first phase, the bids are sorted by some criterion. The algorithms of the family are distinguished by the different criteria they use. Since there are n bids, this phase takes time of the order of n log n. We assume a criterion, that is, a norm is defined and the bids are sorted in decreasing order following this norm. Since we shall have, in Theorem 10.2, to compare the sorted lists of bids of slightly different auctions, we also assume a consistent treatment of ties, that is, bids with equal norms. Formally, we shall assume that no two different bids have the same norm, that is, there are no ties.
—In the second phase, a greedy algorithm generates an allocation. Let L be the list of sorted bids obtained in the first phase. The first bid of L, say b = hs, ai is granted, that is, the set s will be allocated to b and then the algorithm examines each bid of L, in order, and grants it if it does not conflict with any of the bids previously granted. If it does, it denies (i.e., does not grant) the bid. This phase requires time linear in n.
The use of such a greedy scheme is very straightforward and speedy. We shall now discuss its efficiency: how efficient is the allocation generated? The efficiency of the allocation generated depends obviously both on the criterion used in the first phase and on the types of the bidders, or on the distribution with which the bidders are generated. It is clear that, to obtain allocations close to efficiency, one should use a norm that pushes bids that have a good chance to be part of an efficient allocation toward the beginning of the list L. The amount of a bid is a good criterion in this respect: we want bids with higher amounts to have a larger norm than bids with lower
amounts, at least when the bids are for the same set of goods. Similarly, leaving the amount of a bid unchanged but making its bundle a smaller set (inclusion-wise), should also increase the norm. We shall require that changing s to s 0 with s 0 ⊂ s or changing v to v 0 with v 0 > v increase the norm of a bid. Let us call this property bid-monotonicity. This is the only requirement we shall make. Many criteria satisfy it.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

a
.
Definition 7.1. The average amount per good of a bid b = hs, ai is |s|

Sorting the list L by descending average amount per good is a very reasonable idea. But many other possibilities may be considered. Sorting L by descending amounts for example, or, more generally sorting L by a criterion of the form
a/|s|l for some number l, l ≥ 0, possibly depending on k. All such criteria satisfy bid-monotonicity.

[Next verbatim source excerpt]

THEOREM 7.2. The greedy allocation√scheme with norm a/|s|1/2 approximates the optimal allocation within a factor of k.
PROOF. Assume the bids (i.e., √ bidders) are hsi , ai i for i = 1, . . . , n. Let w i = |si |. Our norm is: ri = ai / w i . Let OP be the optimal solution, that is, the

Lean-elaborated paper signature

type:
{Bidder : Type u_1} → {Item : Type u_2} → (Bidder → AppliedModelingLib.Auction.SingleMindedBid Item) → Prop

value:
```
fun {Bidder : Type u_1} {Item : Type u_2} (bids : Bidder → AppliedModelingLib.Auction.SingleMindedBid Item) =>
  LOS02CombinatorialAuctions.SourceCritical.LegalProfile bids ∧
    Function.Injective fun (i : Bidder) => (bids i).sqrtAmountNorm
```

Recorded source-to-declaration screening **Verdict:** matches

Reason: Section 7 formally assumes no two different bids have the same norm; Theorem 7.2 identifies that norm as a/sqrt(|s|). On legal nonempty bids, injectivity of the displayed square-root norm across bidder-indexed bids is exactly this no-ties condition. It is separate from the fixed-priority convention for tied average-value runs.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

LOS02CombinatorialAuctions.SourceCritical.TotalValue

Paper-local declaration:
LOS02CombinatorialAuctions.SourceCritical.TotalValue
Source locator: cited publication, lines 653-664

Verbatim paper-source connection

THEOREM 7.2. The greedy allocation√scheme with norm a/|s|1/2 approximates
the optimal allocation within a factor of k.
PROOF. Assume the bids (i.e.,
√ bidders) are hsi , ai i for i = 1, . . . , n. Let
w i = |si |. Our norm is: ri = ai / w i . Let OP be the optimal solution, that is, the
set ofPbids contained in the optimal solution. The value of the optimal solution is
ai . Let GR be the solution obtained by the greedy allocation and β its
α = i∈OP P
value: β = i∈GR ai . We want to show that:
√
α ≤ β k.
(7)

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will
range over the bidders. Bidders are selfish, but rational, and trying to maximize their
utility in the final outcome. A bidder knows his own utility function (i.e., his type),
but this information is private and neither the auctioneer nor the other players have
access to it. The final result of an auction consists of two elements: an allocation
of the goods and a vector of payments from the bidders to the auctioneer, both of
which are functions of the bidders' declarations (i.e., bids). Formally, we have a
finite set G of k goods and an allocation is a partial function from G to P, that is, a
function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all
goods be allocated. Notice that the allocations produced by the Generalized Vickrey
Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always
total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions
from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types
for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice
that such a type allows for both complementarity and substitutability, but not for
externalities. Since the set 2i does not depend on i, we shall write 2. An element
of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is
also the set of messages that bidder i may send. A bidder may send any element of
2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t
to denote a (true) type, d to denote a message, T or D to denote vectors of n types
and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly
standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G
and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

Our first property requires that the allocation, among single-minded bidders, be
exact, that is, a single-minded bidder either gets exactly the set of goods he desires,
nothing added, or he gets nothing. He never gets only part of what he requested. This
is a very natural property, when dealing with single-minded bidders: the valuation
of the bidder does not increase by giving him part of what he requested instead of
nothing or by giving him more than what he requested instead of just the bundle
he requested.
Exactness Either g j = s or g j = ∅.
In an exact allocation, we shall say that j's bid is granted in the first case, and denied
in the second case. In such a scheme, the allocation may be viewed as a set of bids
(or bidders) that is conflict-free, that is, the s coordinates have pairwise empty
intersections. A GVA, as we defined it, does not in fact always satisfy Exactness.

Lean-elaborated paper signature

```
type:
{Bidder : Type u_1} → {Item : Type u_2} → (Bidder → AppliedModelingLib.Auction.SingleMindedBid Item) → Finset Bidder → ℝ

value:
fun {Bidder : Type u_1} {Item : Type u_2} (bids : Bidder → AppliedModelingLib.Auction.SingleMindedBid Item)
    (selected : Finset Bidder) =>
  AppliedModelingLib.Auction.singleMindedTotalValue bids selected
```

Recorded source-to-declaration screening **Verdict:** matches

Reason: Theorem 7.2 calls the value of a selected solution the sum of the amounts a_i of its selected bids. singleMindedTotalValue is precisely the finite sum of each selected bidder's declared value.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

LOS02CombinatorialAuctions.SourceCritical.Valuation

Paper-local declaration:
LOS02CombinatorialAuctions.SourceCritical.Valuation
Source locator: cited publication, lines 453-455

Verbatim paper-source connection

Definition 5.1. Bidder i is single-minded if and only if there is a set s ⊂ G of
goods and a value v ∈ R+ such that its type t can be described as: t(s 0) = v if s ⊆ s 0
and t(s 0) = 0 otherwise.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will
range over the bidders. Bidders are selfish, but rational, and trying to maximize their
utility in the final outcome. A bidder knows his own utility function (i.e., his type),
but this information is private and neither the auctioneer nor the other players have
access to it. The final result of an auction consists of two elements: an allocation
of the goods and a vector of payments from the bidders to the auctioneer, both of
which are functions of the bidders' declarations (i.e., bids). Formally, we have a
finite set G of k goods and an allocation is a partial function from G to P, that is, a
function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all
goods be allocated. Notice that the allocations produced by the Generalized Vickrey
Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always
total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions
from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types
for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice
that such a type allows for both complementarity and substitutability, but not for
externalities. Since the set 2i does not depend on i, we shall write 2. An element
of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is
also the set of messages that bidder i may send. A bidder may send any element of
2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t
to denote a (true) type, d to denote a message, T or D to denote vectors of n types
and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly
standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G
and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

better approximations if agents are assumed to be single-minded and one parameter. We shall denote by hs, vi the type just described. Note that a
↪ single-minded
bidder enjoys free disposal. We shall assume, in most of this article, that all bidders
are single-minded, that is, there are sets of goods si and nonnegative real numbers
v i such that bidder i is of type hsi , v i i. We shall denote by 6 the set of all singleminded types. The size of the set 6, contrary to the size of 2, is singly
↪ exponential
in k. A string of polynomial size will be enough to code the declarations of the
bidders: it will describe a set of goods and a value. In this setting, we identify bids

[Next verbatim source excerpt]

a
.
Definition 7.1. The average amount per good of a bid b = hs, ai is |s|

[Next verbatim source excerpt]

LEMMA 9.2. In a mechanism that satisfies Exactness and Participation, a
bidder whose bid is denied has utility zero.

Lean-elaborated paper signature

type:
{Item : Type u_1} → AppliedModelingLib.Auction.SingleMindedBid Item → AppliedModelingLib.Auction.Bundle Item → ℝ

value:
```
fun {Item : Type u_1} (bid : AppliedModelingLib.Auction.SingleMindedBid Item)
    (bundle : AppliedModelingLib.Auction.Bundle Item) =>
  bid.valuation bundle
```

Recorded source-to-declaration screening **Verdict:** matches

Reason: Definition 5.1 gives value v exactly when the requested set s is contained in the allocated bundle s' and zero otherwise. The expanded bid.valuation target is this free-disposal formula verbatim in semantic form.

Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

LOS02CombinatorialAuctions.SourceGVA.Admissible

Paper-local declaration:
LOS02CombinatorialAuctions.SourceGVA.Admissible
Source locator: cited publication, lines 205-225

Verbatim paper-source connection

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

Let us denote the bundle obtained by i as:
gi (D) = f (D)−1 (i).

(2)

Lean-elaborated paper signature

```
type:
{Bidder : Type u_1} →
  {Item : Type u_2} →
    [Fintype Bidder] → [Fintype Item] → AppliedModelingLib.Auction.CombinatorialReport Bidder Item → Prop

value:
fun {Bidder : Type u_1} {Item : Type u_2} [Fintype Bidder] [Fintype Item]
    (D : AppliedModelingLib.Auction.CombinatorialReport Bidder Item) =>
  ∀ (i : Bidder) (S : AppliedModelingLib.Auction.Bundle Item), 0 ≤ D i S
```

Recorded source-to-declaration screening **Verdict:** matches

Reason: Section 3 defines both types and messages as nonnegative valuations of every bundle, with finite bidder and good sets. The target universally requires 0 ≤ D i S for every bidder and bundle, exactly that domain.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

LOS02CombinatorialAuctions.SourceGVA.MaximizesFeasible

Paper-local declaration:
LOS02CombinatorialAuctions.SourceGVA.MaximizesFeasible
Source locator: cited publication, lines 282-341

Verbatim paper-source connection

In a GVA, the allocation chosen maximizes the sum of the declared valuations of the bidders, each bidder receives a monetary amount that equals the sum of the declared valuations of all other bidders, and pays the auctioneer the sum of such valuations that would have been obtained if he had not participated in the auction. A way to describe such an auction, in which i does not participate, is to consider the auction in which bidder i declares a zero valuation for all possible bundles. A bidder with zero valuation for all bundles has no influence on the outcome. Formally, given a vector D of declarations, the generalized Vickrey auction defines the allocation and payment policies as follows (notice that a −1 (i) is the bundle allocated to i by allocation a, and that gi is defined in Eq. (2)):
f (D) = argmaxa∈O

n
X
i=1

di (a −1 (i)),

(4)

[form-feed in source extraction]
Approximately Efficient Combinatorial Auctions
n
X

p j (D) = −

n
X

di (gi (D)) +

i=1,i6= j

583

di (gi (Z)),

(5)

i=1,i6= j

where Z i = Di for any i 6= j and Z j (s) = 0 for any bundle s ⊆ G. Since d j (g j (Z)) = 0, we may as well have written:
p j (D) = −

n
X

di (gi (D)) +

i=1,i6= j

n
X

di (gi (Z)).

(6)

i=1

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly

standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G
and payment x is:
u = t(s) − x.

Lean-elaborated paper signature

```
type:
{Bidder : Type u_1} →
  {Item : Type u_2} →
    [Fintype Bidder] →
      [Fintype Item] →
        (AppliedModelingLib.Auction.CombinatorialReport Bidder Item →
           AppliedModelingLib.Auction.BundleAllocation Bidder Item) →
          Prop

value:
fun {Bidder : Type u_1} {Item : Type u_2} [Fintype Bidder] [Fintype Item]
    (alloc :
      AppliedModelingLib.Auction.CombinatorialReport Bidder Item →
        AppliedModelingLib.Auction.BundleAllocation Bidder Item) =>
  ∀ (D : AppliedModelingLib.Auction.CombinatorialReport Bidder Item),
    LOS02CombinatorialAuctions.SourceGVA.Admissible D →
      AppliedModelingLib.Auction.IsFeasibleBundleAllocation (alloc D) Finset.univ ∧
        ∀ (A : AppliedModelingLib.Auction.BundleAllocation Bidder Item),
          AppliedModelingLib.Auction.IsFeasibleBundleAllocation A Finset.univ →
            AppliedModelingLib.Auction.allocationValue D A ≤ AppliedModelingLib.Auction.allocationValue D (alloc D)
```

Recorded source-to-declaration screening

Verdict: matches

Reason: Equation (4) chooses an argmax of the sum of declared values over O, the partial allocations of goods. The target requires the selected allocation to be a feasible partial allocation and to weakly dominate every feasible partial allocation in declared welfare; weak maximality correctly preserves the source's unspecified tie choice.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

LOS02CombinatorialAuctions.SourceGVA.Truthful

Paper-local declaration:
LOS02CombinatorialAuctions.SourceGVA.Truthful
Source locator: cited publication, lines 258-269

Verbatim paper-source connection

Definition 3.2. A mechanism h f, pi is truthful if and only if for every i ∈ P,
t ∈ 2 and any vector D of declarations, if D 0 is the vector obtained from D by
replacing the ith coordinate di by t, then: t(gi (D 0)) − pi (D 0) ≥ t(gi (D)) − pi (D).

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will
range over the bidders. Bidders are selfish, but rational, and trying to maximize their
utility in the final outcome. A bidder knows his own utility function (i.e., his type),
but this information is private and neither the auctioneer nor the other players have
access to it. The final result of an auction consists of two elements: an allocation
of the goods and a vector of payments from the bidders to the auctioneer, both of
which are functions of the bidders' declarations (i.e., bids). Formally, we have a
finite set G of k goods and an allocation is a partial function from G to P, that is, a
function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all
goods be allocated. Notice that the allocations produced by the Generalized Vickrey
Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always
total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions
from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types
for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice
that such a type allows for both complementarity and substitutability, but not for
externalities. Since the set 2i does not depend on i, we shall write 2. An element
of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is
also the set of messages that bidder i may send. A bidder may send any element of
2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t
to denote a (true) type, d to denote a message, T or D to denote vectors of n types
and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly
standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G
and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

Definition 5.1. Bidder i is single-minded if and only if there is a set s ⊂ G of
goods and a value v ∈ R+ such that its type t can be described as: t(s 0) = v if s ⊆ s 0
and t(s 0) = 0 otherwise.

[Next verbatim source excerpt]

The general structure of the properties of interest is that they consider a given
set of single-minded types and vary one of those types. They restrict the changes
that can appear in the allocation or the payments as a result of such a change.
Let declarations be fixed, but arbitrary, for all bidders except j. Consider two
possible declarations for j: hs, vi and hs 0 , v 0 i. Given an allocation scheme f and a
payment scheme p, we shall consider the allocations and payments generated by
both declarations of j. Let gi be the set of goods obtained by bidder i if j declares
hs, vi, and gi0 the set he obtains if j declares hs 0 , v 0 i. Similarly denote by pi and pi0
the payments of i.

[Next verbatim source excerpt]

The game-theoretic solution concept used throughout this article is that of a weakly
dominant strategy, that is, a strategy that is as good as any other for a given player,
no matter what other players do. This is in contrast with the weaker and more
common notion of Nash equilibrium. The particular property we would like to
ensure for our mechanism is that the dominant strategy for each player is to bid
his true valuation; in other words, no bidder can be better off by lying, no matter
how other bidders behave. A mechanism is truthful if no bidder can be better off by
lying, even if other bidders lie. This is a very strong requirement, making for a very
sturdy mechanism.

Lean-elaborated paper signature

```
type:
{Bidder : Type u_1} →
  {Item : Type u_2} →
    [Fintype Bidder] → [Fintype Item] → AppliedModelingLib.Auction.CombinatorialAuction Bidder Item → Prop

value:
fun {Bidder : Type u_1} {Item : Type u_2} [Fintype Bidder] [Fintype Item]
    (M : AppliedModelingLib.Auction.CombinatorialAuction Bidder Item) =>
  ∀ (D : AppliedModelingLib.Auction.CombinatorialReport Bidder Item),
    LOS02CombinatorialAuctions.SourceGVA.Admissible D →
      ∀ (i : Bidder) (report : AppliedModelingLib.Auction.Bundle Item → ℝ),
        (∀ (S : AppliedModelingLib.Auction.Bundle Item), 0 ≤ report S) →
          M.utility D (Function.update D i report) i ≤ M.utility D D i
```

Recorded source-to-declaration screening **Verdict:** matches

Reason: Definition 3.2 fixes other declarations, replaces bidder i's declaration by the true nonnegative type, and requires its quasilinear utility to weakly dominate every nonnegative report. Reparameterizing that source vector by its truthful i-coordinate yields exactly utility D (update D i report) i ≤ utility D D i in the target.

Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

LOS02CombinatorialAuctions.SourceGreedy.averageGreedyMechanism

Paper-local declaration:
LOS02CombinatorialAuctions.SourceGreedy.averageGreedyMechanism
Source locator: cited publication, lines 970-991

Verbatim paper-source connection

We shall now describe the payment mechanism that we propose to be used in conjunction with the greedy allocation of Section 7. The description of the payments is tightly linked with that of the greedy algorithm. The computation of the payment is performed in parallel with the execution of the greedy algorithm and takes time linear in the number of bidders for each payment. On the whole, computing the allocation and the payments takes time at most quadratic in the number of bids. We assume that the criterion used is average amount per good, the adaptation to most other suitable greedy allocations is obvious. Informally, a bidder pays, per good, the average price proposed by the first bid in the list L that is denied because of this bid. Consider a bid j in L. Let c(j) be the average amount per good of j. We shall denote by n(j) the first bid following j (bids are sorted in decreasing order, that is, c(j) ≥ c(n(j))) that has been denied but would have been granted were it not for the presence of j. Assume that such a bid exists. Notice that such a bid necessarily conflicts with j, and therefore:
n(j) = min{i | j < i, s(j) ∩ s(i) 6= ∅, ∀l, l < i, l 6= j, l granted ⇒ s(l) ∩ s(i) = ∅}.
Definition 10.1 Greedy Payment Scheme.
the first phase.

Let L be the sorted list obtained in

— j pays zero if his bid is denied or if there is no bid n(j),
—if there is an n(j) and j's bid hs, vi is granted, he pays |s| × c(n(j)).

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

A single-minded bidder declaring hs, ai, with s ⊆ G and a ∈ R+ will be said to put out a bid b = hs, ai. We shall use s(b) and a(b) to denote the components of b and call a(b) the amount of the bid b. As explained in Section 5, we identify bids and bidders. Two bids b = hs, ai and b0 = hs 0 , a 0 i conflict if s ∩ s 0 6= ∅.
The algorithms we consider execute in two phases.
—In the first phase, the bids are sorted by some criterion. The algorithms of the family are distinguished by the different criteria they use. Since there are n bids, this phase takes time of the order of n log n. We assume a criterion, that is, a norm is defined and the bids are sorted in decreasing order following this norm. Since we shall have, in Theorem 10.2, to compare the sorted lists of bids of slightly different auctions, we also assume a consistent treatment of ties, that is, bids with equal norms. Formally, we shall assume that no two different bids have the same norm, that is, there are no ties.
—In the second phase, a greedy algorithm generates an allocation. Let L be the list of sorted bids obtained in the first phase. The first bid of L, say b = hs, ai is granted, that is, the set s will be allocated to b and then the algorithm examines each bid of L, in order, and grants it if it does not conflict with any of the bids previously granted. If it does, it denies (i.e., does not grant) the bid. This phase requires time linear in n.

[Next verbatim source excerpt]

a
.
Definition 7.1. The average amount per good of a bid b = hs, ai is |s|

Lean-elaborated paper signature

```
type:
{Bidder : Type u_1} →
  {Item : Type u_2} →
```

```
  [Fintype Bidder] → [LinearOrder Bidder] → AppliedModelingLib.Auction.SingleMindedAcceptedMechanism Bidder Item

value:
fun {Bidder : Type u_1} {Item : Type u_2} [Fintype Bidder] [LinearOrder Bidder] =>
  {
    accepted := fun (bids : Bidder → AppliedModelingLib.Auction.SingleMindedBid Item) =>
      AppliedModelingLib.Auction.singleMindedGreedyAcceptedFromOrder bids
        (AppliedModelingLib.Auction.singleMindedAverageOrderOf bids),
    payment := LOS02CombinatorialAuctions.SourceGreedy.averagePayment }
```

Recorded source-to-declaration screening

Verdict: matches

Reason: Sections 7 and 10 sort the complete bid list by decreasing average amount per good, greedily accept a bid exactly when it conflicts with no earlier accepted bid, and pair that run with Definition 10.1 payments. The target uses the complete average order for acceptance and the concrete averagePayment wrapper. Its fixed LinearOrder tie key implements the supplied source-supported consistent-ties convention and is held fixed across reports.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

LOS02CombinatorialAuctions.SourceGreedy.averagePayment

Paper-local declaration:
LOS02CombinatorialAuctions.SourceGreedy.averagePayment
Source locator: cited publication, lines 985-991

Verbatim paper-source connection

Definition 10.1 Greedy Payment Scheme.
the first phase.

Let L be the sorted list obtained in

— j pays zero if his bid is denied or if there is no bid n(j),
—if there is an n(j) and j's bid hs, vi is granted, he pays |s| × c(n(j)).

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

A single-minded bidder declaring hs, ai, with s ⊆ G and a ∈ R+ will be said to put out a bid b = hs, ai. We shall use s(b) and a(b) to denote the components of b and call a(b) the amount of the bid b. As explained in Section 5, we identify bids and bidders. Two bids b = hs, ai and b0 = hs 0 , a 0 i conflict if s ∩ s 0 6= ∅.
The algorithms we consider execute in two phases.
—In the first phase, the bids are sorted by some criterion. The algorithms of the family are distinguished by the different criteria they use. Since there are n bids, this phase takes time of the order of n log n. We assume a criterion, that is, a norm is defined and the bids are sorted in decreasing order following this norm. Since we shall have, in Theorem 10.2, to compare the sorted lists of bids of slightly different auctions, we also assume a consistent treatment of ties, that is, bids with equal norms. Formally, we shall assume that no two different bids have the same norm, that is, there are no ties.
—In the second phase, a greedy algorithm generates an allocation. Let L be the list of sorted bids obtained in the first phase. The first bid of L, say b = hs, ai is granted, that is, the set s will be allocated to b and then the algorithm examines each bid of L, in order, and grants it if it does not conflict with any of the bids previously granted. If it does, it denies (i.e., does not grant) the bid. This phase requires time linear in n.

[Next verbatim source excerpt]

a
.
Definition 7.1. The average amount per good of a bid b = hs, ai is |s|

[Next verbatim source excerpt]

We assume that the criterion used is average amount per good, the adaptation to most other suitable greedy allocations is obvious. Informally, a bidder pays, per good, the average price proposed by the first bid in the list L that is denied because of this bid. Consider a bid j in L. Let c(j) be the average amount per good of j. We shall denote by n(j) the first bid following j (bids are sorted in decreasing order, that is, c(j) ≥ c(n(j))) that has been denied but would have been granted were it not for the presence of j. Assume that such a bid exists. Notice that such a bid necessarily conflicts with j, and therefore:
n(j) = min{i | j < i, s(j) ∩ s(i) 6= ∅, ∀l, l < i, l 6= j, l granted ⇒ s(l) ∩ s(i) = ∅}.

Lean-elaborated paper signature

type:
{Bidder : Type u_1} →
  {Item : Type u_2} →
    [Fintype Bidder] → [LinearOrder Bidder] → (Bidder → AppliedModelingLib.Auction.SingleMindedBid Item) → Bidder → ℝ

value:

```
fun {Bidder : Type u_1} {Item : Type u_2} [Fintype Bidder] [LinearOrder Bidder]
    (bids : Bidder → AppliedModelingLib.Auction.SingleMindedBid Item) (j : Bidder) =>
  AppliedModelingLib.Auction.singleMindedGreedyPaymentFromNextDenied bids
    (AppliedModelingLib.Auction.singleMindedGreedyAcceptedFromOrder bids
      (AppliedModelingLib.Auction.singleMindedAverageOrderOf bids))
    (AppliedModelingLib.Auction.singleMindedGreedyNextDeniedFromOrder bids
      (AppliedModelingLib.Auction.singleMindedAverageOrderOf bids))
    j
```

Recorded source-to-declaration screening

Verdict: matches

Reason: Definition 10.1 charges zero to a denied bid or an accepted bid with no n(j), and otherwise charges |s_j| times the average amount of the first later bid denied because of j. The target supplies one full average-sorted greedy run both to acceptance and to the first qualifying later-bid search, then applies exactly that three-case payment formula; nextDenied is not left arbitrary.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Source claims

Review row 1

Source locator: cited publication, lines 346

Verbatim source input

THEOREM 4.1. The generalized Vickrey auction is a truthful mechanism.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

Definition 3.2. A mechanism h f, pi is truthful if and only if for every i ∈ P, t ∈ 2 and any vector D of declarations, if D 0 is the vector obtained from D by replacing the ith coordinate di by t, then: t(gi (D 0)) − pi (D 0) ≥ t(gi (D)) − pi (D).

[Next verbatim source excerpt]

In a GVA, the allocation chosen maximizes the sum of the declared valuations of the bidders, each bidder receives a monetary amount that equals the sum of the declared valuations of all other bidders, and pays the auctioneer the sum of such valuations that would have been obtained if he had not participated in the auction. A way to describe such an auction, in which i does not participate, is to consider the auction in which bidder i declares a zero valuation for all possible bundles. A bidder with zero valuation for all bundles has no influence on the outcome. Formally, given a vector D of declarations, the generalized Vickrey auction defines the allocation and payment policies as follows (notice that a −1 (i) is the bundle allocated to i by allocation a, and that gi is defined in Eq. (2)):
f (D) = argmaxa∈O

n
X
i=1

di (a −1 (i)),

(4)

[form-feed in source extraction]
Approximately Efficient Combinatorial Auctions
n
X

p j (D) = −

n
X

di (gi (D)) +

i=1,i6= j

583

di (gi (Z)),

(5)

i=1,i6= j

where Z i = Di for any i 6= j and Z j (s) = 0 for any bundle s ⊆ G. Since d j (g j (Z)) = 0, we may as well have written:
p j (D) = −

n

X

di (gi (D)) +

i=1,i6= j

n
X

di (gi (Z)).

(6)

i=1

Expanded PaperInterface specification

```
∀ {Bidder : Type u_1} {Item : Type u_2} [inst : Fintype Bidder] [inst_1 : Fintype Item]
  (alloc :
    AppliedModelingLib.Auction.CombinatorialReport Bidder Item →
      AppliedModelingLib.Auction.BundleAllocation Bidder Item),
  LOS02CombinatorialAuctions.SourceGVA.MaximizesFeasible alloc →
    LOS02CombinatorialAuctions.SourceGVA.Truthful (LOS02CombinatorialAuctions.SourceGVA.auction alloc)
```

Lean proof endpoint (built separately)

LOS02CombinatorialAuctions.PaperInterface.theorem4_1

Recorded source-to-Spec screening **Verdict:** matches

Reason: Pinned Theorem 4.1 says that the GVA is truthful; the pinned context defines truthfulness over every bidder, nonnegative type, and nonnegative alternative declaration, and gives the welfare-maximizing allocation and Clarke-pivot payments. The target assumes exactly the source feasible welfare maximization and applies the source GVA construction, whose ‘Truthful‘ predicate has that full admissible type/report comparison.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 2

Source locator: cited publication, lines 389

Verbatim source input

PROPOSITION 4.2. If j is truthful, his utility u j in the GVA is nonnegative.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

Definition 3.2. A mechanism h f, pi is truthful if and only if for every i ∈ P, t ∈ 2 and any vector D of declarations, if D 0 is the vector obtained from D by replacing the ith coordinate di by t, then: t(gi (D 0)) − pi (D 0) ≥ t(gi (D)) − pi (D).

[Next verbatim source excerpt]

In a GVA, the allocation chosen maximizes the sum of the declared valuations of the bidders, each bidder receives a monetary amount that equals the sum of the declared valuations of all other bidders, and pays the auctioneer the sum of such valuations that would have been obtained if he had not participated in the auction. A way to describe such an auction, in which i does not participate, is to consider the auction in which bidder i declares a zero valuation for all possible bundles. A bidder with zero valuation for all bundles has no influence on the outcome. Formally, given a vector D of declarations, the generalized Vickrey auction defines the allocation and payment policies as follows (notice that a −1 (i) is the bundle allocated to i by allocation a, and that gi is defined in Eq. (2)):
f (D) = argmaxa∈O

n
X
i=1

di (a −1 (i)),

(4)

[form-feed in source extraction]
Approximately Efficient Combinatorial Auctions
n
X

p j (D) = −

n
X

di (gi (D)) +

i=1,i6= j

583

di (gi (Z)),

(5)

i=1,i6= j

where Z i = Di for any i 6= j and Z j (s) = 0 for any bundle s ⊆ G. Since d j (g j (Z)) = 0, we may as well have written:
p j (D) = −

n
X

di (gi (D)) +

i=1,i6= j

n
X

di (gi (Z)).

(6)

i=1

Expanded PaperInterface specification

```
∀ {Bidder : Type u_1} {Item : Type u_2} [inst : Fintype Bidder] [inst_1 : Fintype Item]
  (alloc :
    AppliedModelingLib.Auction.CombinatorialReport Bidder Item →
      AppliedModelingLib.Auction.BundleAllocation Bidder Item),
  LOS02CombinatorialAuctions.SourceGVA.MaximizesFeasible alloc →
    ∀ (D : AppliedModelingLib.Auction.CombinatorialReport Bidder Item),
      LOS02CombinatorialAuctions.SourceGVA.Admissible D →
        ∀ (i : Bidder), 0 ≤ (LOS02CombinatorialAuctions.SourceGVA.auction alloc).utility D D i
```

Lean proof endpoint (built separately)

LOS02CombinatorialAuctions.PaperInterface.proposition4_2

Recorded source-to-Spec screening **Verdict:** matches

Reason: Pinned Proposition 4.2 states that a truthful bidder has nonnegative utility in the GVA. The target assumes the source feasible-welfare maximizer, uses the source Clarke-pivot auction, quantifies every admissible valuation profile and bidder, and asserts exactly nonnegative utility when the profile is both true type and declaration.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 3

Source locator: cited publication, lines 653-654

Verbatim source input

THEOREM 7.2. The greedy allocation√scheme with norm a/|s|1/2 approximates
the optimal allocation within a factor of k.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will
range over the bidders. Bidders are selfish, but rational, and trying to maximize their
utility in the final outcome. A bidder knows his own utility function (i.e., his type),
but this information is private and neither the auctioneer nor the other players have
access to it. The final result of an auction consists of two elements: an allocation
of the goods and a vector of payments from the bidders to the auctioneer, both of
which are functions of the bidders' declarations (i.e., bids). Formally, we have a
finite set G of k goods and an allocation is a partial function from G to P, that is, a
function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all
goods be allocated. Notice that the allocations produced by the Generalized Vickrey
Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always
total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions
from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types
for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice
that such a type allows for both complementarity and substitutability, but not for
externalities. Since the set 2i does not depend on i, we shall write 2. An element
of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is
also the set of messages that bidder i may send. A bidder may send any element of
2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t
to denote a (true) type, d to denote a message, T or D to denote vectors of n types
and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly
standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G
and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

We shall assume that bidders are single-minded and care only about one specific
(bidder-dependent) set of goods. If they do not get this set they value the outcome
at the lowest possible value 0. In other words, our bidders are restricted to one
single bid.
Definition 5.1. Bidder i is single-minded if and only if there is a set s ⊂ G of
goods and a value v ∈ R+ such that its type t can be described as: t(s 0) = v if s ⊆ s 0
and t(s 0) = 0 otherwise.
Note that single-minded bidders are not one parameter agents in the sense
of Archer and Tardos [2001], since they have the freedom of deciding which bundle
they bid for on top on the amount they are willing to pay. In Mu'alem and Nisan
[2002], truthfulness is shown to be attainable for allocation algorithms providing
better approximations if agents are assumed to be single-minded and one parameter. We shall denote by hs, vi the type just described. Note that a
↪ single-minded
bidder enjoys free disposal. We shall assume, in most of this article, that all bidders
are single-minded, that is, there are sets of goods si and nonnegative real numbers
v i such that bidder i is of type hsi , v i i. We shall denote by 6 the set of all singleminded types. The size of the set 6, contrary to the size of 2, is singly
↪ exponential
in k. A string of polynomial size will be enough to code the declarations of the
bidders: it will describe a set of goods and a value. In this setting, we identify bids

[Next verbatim source excerpt]

A single-minded bidder declaring hs, ai, with s ⊆ G and a ∈ R+ will be said to
put out a bid b = hs, ai. We shall use s(b) and a(b) to denote the components of b
and call a(b) the amount of the bid b. As explained in Section 5, we identify bids
and bidders. Two bids b = hs, ai and b0 = hs 0 , a 0 i conflict if s ∩ s 0 6= ∅.
The algorithms we consider execute in two phases.
—In the first phase, the bids are sorted by some criterion. The algorithms of the
family are distinguished by the different criteria they use. Since there are n bids,
this phase takes time of the order of n log n. We assume a criterion, that is, a norm
is defined and the bids are sorted in decreasing order following this norm. Since
we shall have, in Theorem 10.2, to compare the sorted lists of bids of slightly
different auctions, we also assume a consistent treatment of ties, that is, bids
with equal norms. Formally, we shall assume that no two different bids have the
same norm, that is, there are no ties.
—In the second phase, a greedy algorithm generates an allocation. Let L be the
list of sorted bids obtained in the first phase. The first bid of L, say b = hs, ai is
granted, that is, the set s will be allocated to b and then the algorithm examines
each bid of L, in order, and grants it if it does not conflict with any of the bids
previously granted. If it does, it denies (i.e., does not grant) the bid. This phase
requires time linear in n.

[Next verbatim source excerpt]

a
.
Definition 7.1. The average amount per good of a bid b = hs, ai is |s|

[Next verbatim source excerpt]

PROOF. Assume the bids (i.e.,
√ bidders) are hsi , ai i for i = 1, . . . , n. Let
w i = |si |. Our norm is: ri = ai / w i . Let OP be the optimal solution, that is, the
set ofPbids contained in the optimal solution. The value of the optimal solution is
ai . Let GR be the solution obtained by the greedy allocation and β its

α = i∈OP P
value: β = i∈GR ai . We want to show that:
√
α ≤ β k.
(7)

Expanded PaperInterface specification

```
∀ {Bidder : Type u_1} {Item : Type u_2} [Fintype Bidder] [inst : Fintype Item]
  (bids : Bidder → AppliedModelingLib.Auction.SingleMindedBid Item) (order : List Bidder),
  LOS02CombinatorialAuctions.SourceCritical.SqrtNormNoTies bids →
    order.Nodup →
      (∀ (i : Bidder), i ∈ order) →
        AppliedModelingLib.Auction.SingleMindedSqrtNormDescending bids order →
          ∀ (optimal : Finset Bidder),
            AppliedModelingLib.Auction.SingleMindedOptimalAcceptedSet bids optimal →
              AppliedModelingLib.Auction.singleMindedTotalValue bids optimal ≤
                √↑(Fintype.card Item) *
                  AppliedModelingLib.Auction.singleMindedTotalValue bids
                    (AppliedModelingLib.Auction.singleMindedGreedyAcceptedFromOrder bids order)
```

Lean proof endpoint (built separately)

LOS02CombinatorialAuctions.PaperInterface.theorem7_2

Recorded source-to-Spec screening **Verdict:** matches

Reason: Pinned Theorem 7.2 uses the norm ‘a/√|s|‘ and proves the greedy welfare is within a factor ‘√k‘ of optimum. The target explicitly requires legal nonnegative/nonempty bids, the source no-equal-square-root-norm condition, and a complete descending order, then states the equivalent ‘OPT ≤ √|Item| · GREEDY‘ inequality for every feasible optimum.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 4

Source locator: cited publication, lines 757-758

Verbatim source input

8. Greedy Allocation and Clarke’s Payment Scheme Do Not Make a Truthful Mechanism, Even for Single-Minded Bidders

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders’ declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

We shall assume that bidders are single-minded and care only about one specific (bidder-dependent) set of goods. If they do not get this set they value the outcome at the lowest possible value 0. In other words, our bidders are restricted to one single bid.
Definition 5.1. Bidder i is single-minded if and only if there is a set s ⊂ G of goods and a value v ∈ R+ such that its type t can be described as: t(s 0) = v if s ⊆ s 0 and t(s 0) = 0 otherwise.
Note that single-minded bidders are not one parameter agents in the sense of Archer and Tardos [2001], since they have the freedom of deciding which bundle they bid for on top on the amount they are willing to pay. In Mu’alem and Nisan [2002], truthfulness is shown to be attainable for allocation algorithms providing better approximations if agents are assumed to be single-minded and one parameter. We shall denote by hs, vi the type just described. Note that a
↪ single-minded
bidder enjoys free disposal. We shall assume, in most of this article, that all bidders are single-minded, that is, there are sets of goods si and nonnegative real numbers v i such that bidder i is of type hsi , v i i. We shall denote by 6 the set of all singleminded types. The size of the set 6, contrary to the size of 2, is singly
↪ exponential
in k. A string of polynomial size will be enough to code the declarations of the bidders: it will describe a set of goods and a value. In this setting, we identify bids

[Next verbatim source excerpt]

A single-minded bidder declaring hs, ai, with s ⊆ G and a ∈ R+ will be said to put out a bid b = hs, ai. We shall use s(b) and a(b) to denote the components of b and call a(b) the amount of the bid b. As explained in Section 5, we identify bids and bidders. Two bids b = hs, ai and b0 = hs 0 , a 0 i conflict if s ∩ s 0 6= ∅.
The algorithms we consider execute in two phases.
—In the first phase, the bids are sorted by some criterion. The algorithms of the family are distinguished by the different criteria they use. Since there are n bids, this phase takes time of the order of n log n. We assume a criterion, that is, a norm is defined and the bids are sorted in decreasing order following this norm. Since we shall have, in Theorem 10.2, to compare the sorted lists of bids of slightly different auctions, we also assume a consistent treatment of ties, that is, bids with equal norms. Formally, we shall assume that no two different bids have the same norm, that is, there are no ties.
—In the second phase, a greedy algorithm generates an allocation. Let L be the list of sorted bids obtained in the first phase. The first bid of L, say b = hs, ai is granted, that is, the set s will be allocated to b and then the algorithm examines each bid of L, in order, and grants it if it does not conflict with any of the bids previously granted. If it does, it denies (i.e., does not grant) the bid. This phase requires time linear in n.

[Next verbatim source excerpt]

a
.
Definition 7.1. The average amount per good of a bid b = hs, ai is |s|

[Next verbatim source excerpt]

In a GVA, the allocation chosen maximizes the sum of the declared valuations of the bidders, each bidder receives a monetary amount that equals the sum of the declared valuations of all other bidders, and pays the auctioneer the sum of such valuations that would have been obtained if he had not participated in the auction. A way to describe such an auction, in which i does not participate, is to consider

the auction in which bidder i declares a zero valuation for all possible bundles. A bidder with zero valuation for all bundles has no influence on the outcome. Formally, given a vector D of declarations, the generalized Vickrey auction defines the allocation and payment policies as follows (notice that a −1 (i) is the bundle allocated to i by allocation a, and that gi is defined in Eq. (2)):
f (D) = argmaxa∈O

n
X
i=1

di (a −1 (i)),

(4)

[form-feed in source extraction]
Approximately Efficient Combinatorial Auctions
n
X

p j (D) = −

n
X

di (gi (D)) +

i=1,i6= j

583

di (gi (Z)),

(5)

i=1,i6= j

where Z i = Di for any i 6= j and Z j (s) = 0 for any bundle s ⊆ G. Since d j (g j (Z)) = 0, we may as well have written:
p j (D) = −

n
X

di (gi (D)) +

i=1,i6= j

n
X

di (gi (Z)).

(6)

i=1

[Next verbatim source excerpt]

norms. In the sequel, all examples will use the average amount per good criterion but it is not difficult to find similar examples for any criterion of the form a/|s|l .

[Next verbatim source excerpt]

Example 8.1. As in Example 7.3, there are two goods a and b and three bidders Red, Green, and Blue. Red bids 10 for a, Green bids 19 for the set {a, b} and Blue bids 8 for b. The greedy algorithm grants Red's and Blue's bids and denies Green's bid, that is, f (D)(a) = Red and f (D)(b) = Blue. We shall now compute Red's payment. For this allocation, we have the following declared valuations: v Blue = 8 and v Green = 0. If Red had bid zero, the greedy algorithm would have granted Green's bid and denied Blue's bid. Therefore, the allocation f (Z) is defined by: f (Z)(a) = f (Z)(b) = Green, where v Blue = 0 and v Green = 19. Clarke's payment scheme gives to Red: 8 − 0 for Blue and 0 − 19 for Green, that is, Red pays 11.

Expanded PaperInterface specification

```
LOS02CombinatorialAuctions.NegativeResults.section8Witness ∧
  ¬LOS02CombinatorialAuctions.NegativeResults.greedyClarke.TruthfulOn
      AppliedModelingLib.Auction.SingleMindedAcceptedMechanism.NonnegativeNonemptyProfile
```

Lean proof endpoint (built separately)

```
LOS02CombinatorialAuctions.PaperInterface.section8
```

Recorded source-to-Spec screening **Verdict:** matches

Reason: The pinned Section 8 counterexample has Red bid 10 for 'a', Green 19 for '{a,b}', Blue 8 for 'b', and compares Red's truthful run with a zero report; it computes the Clarke payment of 11 in the truthful run. The target records the corresponding source witness utilities '-1' and '0' and concludes nontruthfulness on the legal single-minded domain. These explicit witness bids are nonempty, and the approved fixed tie rule is used consistently.
Reviewer check: □ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 5

Source locator: cited publication, lines 878-884

Verbatim source input

LEMMA 9.1. In a mechanism that satisfies Exactness and Monotonicity, given
a bidder j, a set s of goods and declarations for all other bidders, there exists a
critical value v c such that
∀v, v < v c ⇒ g j = ∅,
∀v, v > v c ⇒ g j = s,
We allow v c to be infinite if f (As,v)−1 (j) = ∅ for every v. Note that we do not
know whether j's bid is granted or not in case v = v c .

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will
range over the bidders. Bidders are selfish, but rational, and trying to maximize their
utility in the final outcome. A bidder knows his own utility function (i.e., his type),
but this information is private and neither the auctioneer nor the other players have
access to it. The final result of an auction consists of two elements: an allocation
of the goods and a vector of payments from the bidders to the auctioneer, both of
which are functions of the bidders' declarations (i.e., bids). Formally, we have a
finite set G of k goods and an allocation is a partial function from G to P, that is, a
function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all
goods be allocated. Notice that the allocations produced by the Generalized Vickrey
Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always
total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions
from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types
for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice
that such a type allows for both complementarity and substitutability, but not for
externalities. Since the set 2i does not depend on i, we shall write 2. An element
of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is
also the set of messages that bidder i may send. A bidder may send any element of
2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t
to denote a (true) type, d to denote a message, T or D to denote vectors of n types
and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly
standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G
and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

We shall assume that bidders are single-minded and care only about one specific
(bidder-dependent) set of goods. If they do not get this set they value the outcome
at the lowest possible value 0. In other words, our bidders are restricted to one
single bid.
Definition 5.1. Bidder i is single-minded if and only if there is a set s ⊂ G of
goods and a value v ∈ R+ such that its type t can be described as: t(s 0) = v if s ⊆ s 0
and t(s 0) = 0 otherwise.
Note that single-minded bidders are not one parameter agents in the sense
of Archer and Tardos [2001], since they have the freedom of deciding which bundle
they bid for on top on the amount they are willing to pay. In Mu'alem and Nisan
[2002], truthfulness is shown to be attainable for allocation algorithms providing
better approximations if agents are assumed to be single-minded and one parameter. We shall denote by hs, vi the type just described. Note that a
↪ single-minded
bidder enjoys free disposal. We shall assume, in most of this article, that all bidders
are single-minded, that is, there are sets of goods si and nonnegative real numbers
v i such that bidder i is of type hsi , v i i. We shall denote by 6 the set of all singleminded types. The size of the set 6, contrary to the size of 2, is singly
↪ exponential
in k. A string of polynomial size will be enough to code the declarations of the
bidders: it will describe a set of goods and a value. In this setting, we identify bids

[Next verbatim source excerpt]

The general structure of the properties of interest is that they consider a given
set of single-minded types and vary one of those types. They restrict the changes
that can appear in the allocation or the payments as a result of such a change.
Let declarations be fixed, but arbitrary, for all bidders except j. Consider two
possible declarations for j: hs, vi and hs 0 , v 0 i. Given an allocation scheme f and a
payment scheme p, we shall consider the allocations and payments generated by
both declarations of j. Let gi be the set of goods obtained by bidder i if j declares
hs, vi, and gi0 the set he obtains if j declares hs 0 , v 0 i. Similarly denote by pi and pi0
the payments of i.
Our first property requires that the allocation, among single-minded bidders, be
exact, that is, a single-minded bidder either gets exactly the set of goods he desires,
nothing added, or he gets nothing. He never gets only part of what he requested. This
is a very natural property, when dealing with single-minded bidders: the valuation
of the bidder does not increase by giving him part of what he requested instead of
nothing or by giving him more than what he requested instead of just the bundle
he requested.
Exactness Either g j = s or g j = ∅.
In an exact allocation, we shall say that j's bid is granted in the first case, and denied
in the second case. In such a scheme, the allocation may be viewed as a set of bids
(or bidders) that is conflict-free, that is, the s coordinates have pairwise empty

[Next verbatim source excerpt]

Our next property, Monotonicity, also concerns only the allocation scheme. It
requires that, if j's bid is granted if he declares hs, vi, it is also granted if he
declares hs 0 , v 0 i for any s 0 ⊆ s, v 0 ≥ v. In other words, proposing more money for
fewer goods cannot cause a bidder to lose his bid. It follows that, similarly, offering
less money for more goods cannot cause a lost bid to win. Formally:

Monotonicity s ⊆ g j ,

s 0 ⊆ s,

v 0 ≥ v ⇒ s 0 ⊆ g 0j .

Expanded PaperInterface specification

```
∀ {Bidder : Type u_1} {Item : Type u_2} [Fintype Bidder] [Fintype Item]
  (M : AppliedModelingLib.Auction.SingleMindedAcceptedMechanism Bidder Item),
  LOS02CombinatorialAuctions.SourceCritical.Feasible M →
    M.MonotonicityOn AppliedModelingLib.Auction.SingleMindedAcceptedMechanism.NonnegativeNonemptyProfile →
      ∀ (D : Bidder → AppliedModelingLib.Auction.SingleMindedBid Item),
        AppliedModelingLib.Auction.SingleMindedAcceptedMechanism.NonnegativeNonemptyProfile D →
          ∀ (i : Bidder) (s : AppliedModelingLib.Auction.Bundle Item),
            Finset.Nonempty s →
              (∃ (c : ℝ),
                  0 ≤ c ∧
                    (∀ (v : ℝ), 0 ≤ v → v < c → i ∉ M.accepted (Function.update D i { desired := s, value := v })) ∧
                      ∀ (v : ℝ), 0 ≤ v → c < v → i ∈ M.accepted (Function.update D i { desired := s, value := v })) ∨
                ∀ (v : ℝ), 0 ≤ v → i ∉ M.accepted (Function.update D i { desired := s, value := v })
```

Lean proof endpoint (built separately)

LOS02CombinatorialAuctions.PaperInterface.lemma9_1

Recorded source-to-Spec screening **Verdict:** matches

Reason: Pinned Lemma 9.1 derives a critical value from Exactness and Monotonicity: values below it are denied, values above it are granted, equality is unspecified, and an always-denied infinite branch is permitted. The target encodes exactness through the accepted-set model plus ‘Feasible‘, restricts bids to the approved nonnegative/nonempty domain, gives the same finite nonnegative threshold versus always-denied alternative, and leaves equality unconstrained.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 6

Source locator: cited publication, lines 924-925

Verbatim source input

LEMMA 9.2. In a mechanism that satisfies Exactness and Participation, a
bidder whose bid is denied has utility zero.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will
range over the bidders. Bidders are selfish, but rational, and trying to maximize their
utility in the final outcome. A bidder knows his own utility function (i.e., his type),
but this information is private and neither the auctioneer nor the other players have
access to it. The final result of an auction consists of two elements: an allocation
of the goods and a vector of payments from the bidders to the auctioneer, both of
which are functions of the bidders' declarations (i.e., bids). Formally, we have a
finite set G of k goods and an allocation is a partial function from G to P, that is, a
function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all
goods be allocated. Notice that the allocations produced by the Generalized Vickrey
Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always
total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions
from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types
for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice
that such a type allows for both complementarity and substitutability, but not for
externalities. Since the set 2i does not depend on i, we shall write 2. An element
of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is
also the set of messages that bidder i may send. A bidder may send any element of
2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t
to denote a (true) type, d to denote a message, T or D to denote vectors of n types
and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly
standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G
and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

We shall assume that bidders are single-minded and care only about one specific
(bidder-dependent) set of goods. If they do not get this set they value the outcome
at the lowest possible value 0. In other words, our bidders are restricted to one
single bid.
Definition 5.1. Bidder i is single-minded if and only if there is a set s ⊂ G of
goods and a value v ∈ R+ such that its type t can be described as: t(s 0) = v if s ⊆ s 0
and t(s 0) = 0 otherwise.
Note that single-minded bidders are not one parameter agents in the sense
of Archer and Tardos [2001], since they have the freedom of deciding which bundle
they bid for on top on the amount they are willing to pay. In Mu'alem and Nisan
[2002], truthfulness is shown to be attainable for allocation algorithms providing
better approximations if agents are assumed to be single-minded and one parameter. We shall denote by hs, vi the type just described. Note that a
↪ single-minded
bidder enjoys free disposal. We shall assume, in most of this article, that all bidders
are single-minded, that is, there are sets of goods si and nonnegative real numbers
v i such that bidder i is of type hsi , v i i. We shall denote by 6 the set of all singleminded types. The size of the set 6, contrary to the size of 2, is singly
↪ exponential
in k. A string of polynomial size will be enough to code the declarations of the
bidders: it will describe a set of goods and a value. In this setting, we identify bids

[Next verbatim source excerpt]

The general structure of the properties of interest is that they consider a given
set of single-minded types and vary one of those types. They restrict the changes
that can appear in the allocation or the payments as a result of such a change.
Let declarations be fixed, but arbitrary, for all bidders except j. Consider two
possible declarations for j: hs, vi and hs 0 , v 0 i. Given an allocation scheme f and a
payment scheme p, we shall consider the allocations and payments generated by
both declarations of j. Let gi be the set of goods obtained by bidder i if j declares
hs, vi, and gi0 the set he obtains if j declares hs 0 , v 0 i. Similarly denote by pi and pi0
the payments of i.
Our first property requires that the allocation, among single-minded bidders, be
exact, that is, a single-minded bidder either gets exactly the set of goods he desires,
nothing added, or he gets nothing. He never gets only part of what he requested. This
is a very natural property, when dealing with single-minded bidders: the valuation
of the bidder does not increase by giving him part of what he requested instead of
nothing or by giving him more than what he requested instead of just the bundle
he requested.
Exactness Either g j = s or g j = ∅.
In an exact allocation, we shall say that j's bid is granted in the first case, and denied
in the second case. In such a scheme, the allocation may be viewed as a set of bids
(or bidders) that is conflict-free, that is, the s coordinates have pairwise empty

[Next verbatim source excerpt]

Our next property, Monotonicity, also concerns only the allocation scheme. It
requires that, if j's bid is granted if he declares hs, vi, it is also granted if he
declares hs 0 , v 0 i for any s 0 ⊆ s, v 0 ≥ v. In other words, proposing more money for
fewer goods cannot cause a bidder to lose his bid. It follows that, similarly, offering
less money for more goods cannot cause a lost bid to win. Formally:
Monotonicity s ⊆ g j ,

s 0 ⊆ s,

v 0 ≥ v ⇒ s 0 ⊆ g 0j .

[Next verbatim source excerpt]

Our last property concerns the payment scheme. Together with Critical, it implies that the utility of no truthful bidder is negative. It concerns unsatisfied bidders, bids that are denied. We require that an unsatisfied bidder pay zero. The utility of an unsatisfied bidder is then zero. This is simply tuning the utility scales of the different bidders, or, ensuring that bidders may not lose by participating in the auction. Participation

s 6⊆ g j ⇒ p j = 0

Expanded PaperInterface specification

```
∀ {Bidder : Type u_1} {Item : Type u_2} [Fintype Bidder] [Fintype Item]
  (M : AppliedModelingLib.Auction.SingleMindedAcceptedMechanism Bidder Item),
  LOS02CombinatorialAuctions.SourceCritical.Feasible M →
    LOS02CombinatorialAuctions.SourceCritical.Participation M →
      ∀ (T D : Bidder → AppliedModelingLib.Auction.SingleMindedBid Item),
        AppliedModelingLib.Auction.SingleMindedAcceptedMechanism.NonnegativeNonemptyProfile T →
          AppliedModelingLib.Auction.SingleMindedAcceptedMechanism.NonnegativeNonemptyProfile D →
            ∀ i ∉ M.accepted D, M.utility T D i = 0
```

Lean proof endpoint (built separately)

LOS02CombinatorialAuctions.PaperInterface.lemma9_2

Recorded source-to-Spec screening **Verdict:** matches

Reason: Pinned Lemma 9.2 says Exactness and Participation make a denied bidder's utility zero. The target's accepted-set model and 'Feasible' supply exactness, 'Participation' makes the denied report's payment zero on the legal domain, and the explicit legal true-type/report profiles make the resulting cross-profile utility exactly zero.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 7

Source locator: cited publication, lines 928

Verbatim source input

LEMMA 9.3. In a mechanism that satisfies Exactness, Monotonicity, Participation and Critical a truthful bidder's utility is nonnegative.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will
range over the bidders. Bidders are selfish, but rational, and trying to maximize their
utility in the final outcome. A bidder knows his own utility function (i.e., his type),
but this information is private and neither the auctioneer nor the other players have
access to it. The final result of an auction consists of two elements: an allocation
of the goods and a vector of payments from the bidders to the auctioneer, both of
which are functions of the bidders' declarations (i.e., bids). Formally, we have a
finite set G of k goods and an allocation is a partial function from G to P, that is, a
function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all
goods be allocated. Notice that the allocations produced by the Generalized Vickrey
Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always
total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions
from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types
for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice
that such a type allows for both complementarity and substitutability, but not for
externalities. Since the set 2i does not depend on i, we shall write 2. An element
of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is
also the set of messages that bidder i may send. A bidder may send any element of
2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t
to denote a (true) type, d to denote a message, T or D to denote vectors of n types
and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly
standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G
and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

We shall assume that bidders are single-minded and care only about one specific
(bidder-dependent) set of goods. If they do not get this set they value the outcome
at the lowest possible value 0. In other words, our bidders are restricted to one
single bid.
Definition 5.1. Bidder i is single-minded if and only if there is a set s ⊂ G of
goods and a value v ∈ R+ such that its type t can be described as: t(s 0) = v if s ⊆ s 0
and t(s 0) = 0 otherwise.
Note that single-minded bidders are not one parameter agents in the sense
of Archer and Tardos [2001], since they have the freedom of deciding which bundle
they bid for on top on the amount they are willing to pay. In Mu'alem and Nisan
[2002], truthfulness is shown to be attainable for allocation algorithms providing
better approximations if agents are assumed to be single-minded and one parameter. We shall denote by hs, vi the type just described. Note that a
↪ single-minded
bidder enjoys free disposal. We shall assume, in most of this article, that all bidders
are single-minded, that is, there are sets of goods si and nonnegative real numbers
v i such that bidder i is of type hsi , v i i. We shall denote by 6 the set of all singleminded types. The size of the set 6, contrary to the size of 2, is singly
↪ exponential
in k. A string of polynomial size will be enough to code the declarations of the
bidders: it will describe a set of goods and a value. In this setting, we identify bids

[Next verbatim source excerpt]

The general structure of the properties of interest is that they consider a given
set of single-minded types and vary one of those types. They restrict the changes
that can appear in the allocation or the payments as a result of such a change.
Let declarations be fixed, but arbitrary, for all bidders except j. Consider two
possible declarations for j: hs, vi and hs 0 , v 0 i. Given an allocation scheme f and a
payment scheme p, we shall consider the allocations and payments generated by
both declarations of j. Let gi be the set of goods obtained by bidder i if j declares
hs, vi, and gi0 the set he obtains if j declares hs 0 , v 0 i. Similarly denote by pi and pi0
the payments of i.
Our first property requires that the allocation, among single-minded bidders, be
exact, that is, a single-minded bidder either gets exactly the set of goods he desires,
nothing added, or he gets nothing. He never gets only part of what he requested. This
is a very natural property, when dealing with single-minded bidders: the valuation
of the bidder does not increase by giving him part of what he requested instead of
nothing or by giving him more than what he requested instead of just the bundle
he requested.
Exactness Either g j = s or g j = ∅.
In an exact allocation, we shall say that j's bid is granted in the first case, and denied
in the second case. In such a scheme, the allocation may be viewed as a set of bids
(or bidders) that is conflict-free, that is, the s coordinates have pairwise empty

[Next verbatim source excerpt]

Our next property, Monotonicity, also concerns only the allocation scheme. It
requires that, if j's bid is granted if he declares hs, vi, it is also granted if he
declares hs 0 , v 0 i for any s 0 ⊆ s, v 0 ≥ v. In other words, proposing more money for
fewer goods cannot cause a bidder to lose his bid. It follows that, similarly, offering
less money for more goods cannot cause a lost bid to win. Formally:
Monotonicity s ⊆ g j ,

s 0 ⊆ s,

v 0 ≥ v ⇒ s 0 ⊆ g 0j .

[Next verbatim source excerpt]

LEMMA 9.1. In a mechanism that satisfies Exactness and Monotonicity, given a bidder j, a set s of goods and declarations for all other bidders, there exists a critical value v c such that
∀v, v < v c ⇒ g j = ∅,
∀v, v > v c ⇒ g j = s,
We allow v c to be infinite if f (As,v)−1 (j) = ∅ for every v. Note that we do not know whether j's bid is granted or not in case v = v c .

[Next verbatim source excerpt]

Our third property deals with a satisfied bidder: a satisfied bidder pays exactly the critical value of Lemma 9.1, that is, the lowest value he could have declared and still be allocated the goods he desires.
Critical

s ⊆ g j ⇒ p j = vc

Notice that Critical says, first, that the payment for a bid that is granted does not depend on the amount of the bid, it depends only on the other bids. Then it says that it is exactly equal to the critical value below which the bid would have lost.

[Next verbatim source excerpt]

Our last property concerns the payment scheme. Together with Critical, it implies that the utility of no truthful bidder is negative. It concerns unsatisfied bidders, bids that are denied. We require that an unsatisfied bidder pay zero. The utility of an unsatisfied bidder is then zero. This is simply tuning the utility scales of the different bidders, or, ensuring that bidders may not lose by participating in the auction.
Participation

s 6⊆ g j ⇒ p j = 0

Expanded PaperInterface specification

```
∀ {Bidder : Type u_1} {Item : Type u_2} [Fintype Bidder] [Fintype Item]
  (M : AppliedModelingLib.Auction.SingleMindedAcceptedMechanism Bidder Item),
  LOS02CombinatorialAuctions.SourceCritical.Feasible M →
    M.MonotonicityOn AppliedModelingLib.Auction.SingleMindedAcceptedMechanism.NonnegativeNonemptyProfile →
      LOS02CombinatorialAuctions.SourceCritical.Participation M →
        ∀ (_C : M.NonnegativeCriticalValueWithInfinityCertificate)
          (T : Bidder → AppliedModelingLib.Auction.SingleMindedBid Item),
          AppliedModelingLib.Auction.SingleMindedAcceptedMechanism.NonnegativeNonemptyProfile T →
            ∀ (i : Bidder), 0 ≤ M.utility T T i
```

Lean proof endpoint (built separately)

LOS02CombinatorialAuctions.PaperInterface.lemma9_3

Recorded source-to-Spec screening **Verdict:** matches

Reason: Pinned Lemma 9.3 assumes Exactness, Monotonicity, Participation, and Critical and concludes nonnegative truthful utility. The target has the accepted-set exactness/feasibility condition, monotonicity and participation on the approved legal domain, and the nonnegative finite-or-infinite critical-value certificate, then concludes '0 ≤ utility T T i' for every legal truthful profile.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

Review row 8

Source locator: cited publication, lines 935-936

Verbatim source input

LEMMA 9.4. In a mechanism that satisfies Exactness, Monotonicity, Participation and Critical, a bidder j of type hs, vi is never better off declaring hs, v 0 i for some v 0 6= v than by being truthful.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

We shall assume that bidders are single-minded and care only about one specific (bidder-dependent) set of goods. If they do not get this set they value the outcome at the lowest possible value 0. In other words, our bidders are restricted to one single bid.
Definition 5.1. Bidder i is single-minded if and only if there is a set s ⊂ G of goods and a value v ∈ R+ such that its type t can be described as: t(s 0) = v if s ⊆ s 0 and t(s 0) = 0 otherwise.
Note that single-minded bidders are not one parameter agents in the sense of Archer and Tardos [2001], since they have the freedom of deciding which bundle they bid for on top on the amount they are willing to pay. In Mu'alem and Nisan [2002], truthfulness is shown to be attainable for allocation algorithms providing better approximations if agents are assumed to be single-minded and one parameter. We shall denote by hs, vi the type just described. Note that a
↪ single-minded
bidder enjoys free disposal. We shall assume, in most of this article, that all bidders are single-minded, that is, there are sets of goods si and nonnegative real numbers v i such that bidder i is of type hsi , v i i. We shall denote by 6 the set of all singleminded types. The size of the set 6, contrary to the size of 2, is singly
↪ exponential
in k. A string of polynomial size will be enough to code the declarations of the bidders: it will describe a set of goods and a value. In this setting, we identify bids

[Next verbatim source excerpt]

The general structure of the properties of interest is that they consider a given set of single-minded types and vary one of those types. They restrict the changes that can appear in the allocation or the payments as a result of such a change.
Let declarations be fixed, but arbitrary, for all bidders except j. Consider two possible declarations for j: hs, vi and hs 0 , v 0 i. Given an allocation scheme f and a payment scheme p, we shall consider the allocations and payments generated by both declarations of j. Let gi be the set of goods obtained by bidder i if j declares hs, vi, and gi0 the set he obtains if j declares hs 0 , v 0 i. Similarly denote by pi and pi0 the payments of i.
Our first property requires that the allocation, among single-minded bidders, be exact, that is, a single-minded bidder either gets exactly the set of goods he desires, nothing added, or he gets nothing. He never gets only part of what he requested. This is a very natural property, when dealing with single-minded bidders: the valuation of the bidder does not increase by giving him part of what he requested instead of nothing or by giving him more than what he requested instead of just the bundle he requested.
Exactness Either g j = s or g j = ∅.
In an exact allocation, we shall say that j's bid is granted in the first case, and denied in the second case. In such a scheme, the allocation may be viewed as a set of bids (or bidders) that is conflict-free, that is, the s coordinates have pairwise empty

[Next verbatim source excerpt]

Our next property, Monotonicity, also concerns only the allocation scheme. It requires that, if j's bid is granted if he declares hs, vi, it is also granted if he declares hs 0 , v 0 i for any s 0 ⊆ s, v 0 ≥ v. In other words, proposing more money for fewer goods cannot cause a bidder to lose his bid. It follows that, similarly, offering less money for more goods cannot cause a lost bid to win. Formally:
Monotonicity s ⊆ g j ,

s 0 ⊆ s,

v 0 ≥ v ⇒ s 0 ⊆ g 0j .

[Next verbatim source excerpt]

LEMMA 9.1. In a mechanism that satisfies Exactness and Monotonicity, given a bidder j, a set s of goods and declarations for all other bidders, there exists a critical value v c such that
∀v, v < v c ⇒ g j = ∅,
∀v, v > v c ⇒ g j = s,
We allow v c to be infinite if f (As,v)−1 (j) = ∅ for every v. Note that we do not know whether j's bid is granted or not in case v = v c .

[Next verbatim source excerpt]

Our third property deals with a satisfied bidder: a satisfied bidder pays exactly the critical value of Lemma 9.1, that is, the lowest value he could have declared and still be allocated the goods he desires.
Critical

s ⊆ g j ⇒ p j = vc

Notice that Critical says, first, that the payment for a bid that is granted does not depend on the amount of the bid, it depends only on the other bids. Then it says that it is exactly equal to the critical value below which the bid would have lost.

[Next verbatim source excerpt]

Our last property concerns the payment scheme. Together with Critical, it implies that the utility of no truthful bidder is negative. It concerns unsatisfied bidders, bids that are denied. We require that an unsatisfied bidder pay zero. The utility of an unsatisfied bidder is then zero. This is simply tuning the utility scales of the different bidders, or, ensuring that bidders may not lose by participating in the auction.
Participation

s 6⊆ g j ⇒ p j = 0

Expanded PaperInterface specification

```
∀ {Bidder : Type u_1} {Item : Type u_2} [Fintype Bidder] [Fintype Item]
  (M : AppliedModelingLib.Auction.SingleMindedAcceptedMechanism Bidder Item),
  LOS02CombinatorialAuctions.SourceCritical.Feasible M →
    M.MonotonicityOn AppliedModelingLib.Auction.SingleMindedAcceptedMechanism.NonnegativeNonemptyProfile →
      LOS02CombinatorialAuctions.SourceCritical.Participation M →
        ∀ (_C : M.NonnegativeCriticalValueWithInfinityCertificate)
          (T : Bidder → AppliedModelingLib.Auction.SingleMindedBid Item),
          AppliedModelingLib.Auction.SingleMindedAcceptedMechanism.NonnegativeNonemptyProfile T →
            ∀ (i : Bidder) (v : ℝ),
              0 ≤ v → M.utility T (Function.update T i { desired := (T i).desired, value := v }) i ≤ M.utility T T i
```

Lean proof endpoint (built separately)

LOS02CombinatorialAuctions.PaperInterface.lemma9_4

Recorded source-to-Spec screening **Verdict:** matches

Reason: Pinned Lemma 9.4 rules out a beneficial value-only declaration ‘⟨s,v'⟩‘ by a bidder of type ‘⟨s,v⟩‘, under Exactness, Monotonicity, Participation, and Critical. The target has precisely those modeled conditions and compares the utility from updating only the declared value with truthful utility for every nonnegative ‘v''; its inclusion of ‘v' = v‘ merely includes the trivial truthful case and does not alter the source claim.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 9

Source locator: cited publication, lines 946-948

Verbatim source input

LEMMA 9.5. In a mechanism that satisfies Exactness, Monotonicity and Critical, a bidder j declaring type hs, vi whose bid is granted, that is, g j = s, pays a price p j that is at least the price p 0j that he would have paid had he declared his type as hs 0 , vi for any s 0 ⊆ s.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will range over the bidders. Bidders are selfish, but rational, and trying to maximize their utility in the final outcome. A bidder knows his own utility function (i.e., his type), but this information is private and neither the auctioneer nor the other players have access to it. The final result of an auction consists of two elements: an allocation of the goods and a vector of payments from the bidders to the auctioneer, both of which are functions of the bidders' declarations (i.e., bids). Formally, we have a finite set G of k goods and an allocation is a partial function from G to P, that is, a function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all goods be allocated. Notice that the allocations produced by the Generalized Vickrey Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice that such a type allows for both complementarity and substitutability, but not for externalities. Since the set 2i does not depend on i, we shall write 2. An element of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is also the set of messages that bidder i may send. A bidder may send any element of 2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t to denote a (true) type, d to denote a message, T or D to denote vectors of n types and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

We shall assume that bidders are single-minded and care only about one specific (bidder-dependent) set of goods. If they do not get this set they value the outcome at the lowest possible value 0. In other words, our bidders are restricted to one single bid.
Definition 5.1. Bidder i is single-minded if and only if there is a set s ⊂ G of goods and a value v ∈ R+ such that its type t can be described as: t(s 0) = v if s ⊆ s 0 and t(s 0) = 0 otherwise.
Note that single-minded bidders are not one parameter agents in the sense of Archer and Tardos [2001], since they have the freedom of deciding which bundle they bid for on top on the amount they are willing to pay. In Mu'alem and Nisan [2002], truthfulness is shown to be attainable for allocation algorithms providing better approximations if agents are assumed to be single-minded and one parameter. We shall denote by hs, vi the type just described. Note that a
↪ single-minded
bidder enjoys free disposal. We shall assume, in most of this article, that all bidders are single-minded, that is, there are sets of goods si and nonnegative real numbers v i such that bidder i is of type hsi , v i i. We shall denote by 6 the set of all singleminded types. The size of the set 6, contrary to the size of 2, is singly
↪ exponential
in k. A string of polynomial size will be enough to code the declarations of the bidders: it will describe a set of goods and a value. In this setting, we identify bids

[Next verbatim source excerpt]

The general structure of the properties of interest is that they consider a given set of single-minded types and vary one of those types. They restrict the changes that can appear in the allocation or the payments as a result of such a change.
Let declarations be fixed, but arbitrary, for all bidders except j. Consider two possible declarations for j: hs, vi and hs 0 , v 0 i. Given an allocation scheme f and a payment scheme p, we shall consider the allocations and payments generated by both declarations of j. Let gi be the set of goods obtained by bidder i if j declares hs, vi, and gi0 the set he obtains if j declares hs 0 , v 0 i. Similarly denote by pi and pi0 the payments of i.
Our first property requires that the allocation, among single-minded bidders, be exact, that is, a single-minded bidder either gets exactly the set of goods he desires, nothing added, or he gets nothing. He never gets only part of what he requested. This is a very natural property, when dealing with single-minded bidders: the valuation of the bidder does not increase by giving him part of what he requested instead of nothing or by giving him more than what he requested instead of just the bundle he requested.
Exactness Either g j = s or g j = ∅.
In an exact allocation, we shall say that j's bid is granted in the first case, and denied in the second case. In such a scheme, the allocation may be viewed as a set of bids (or bidders) that is conflict-free, that is, the s coordinates have pairwise empty

[Next verbatim source excerpt]

Our next property, Monotonicity, also concerns only the allocation scheme. It requires that, if j's bid is granted if he declares hs, vi, it is also granted if he declares hs 0 , v 0 i for any s 0 ⊆ s, v 0 ≥ v. In other words, proposing more money for fewer goods cannot cause a bidder to lose his bid. It follows that, similarly, offering less money for more goods cannot cause a lost bid to win. Formally:
Monotonicity s ⊆ g j ,

s 0 ⊆ s,

v 0 ≥ v ⇒ s 0 ⊆ g 0j .

[Next verbatim source excerpt]

LEMMA 9.1. In a mechanism that satisfies Exactness and Monotonicity, given
a bidder j, a set s of goods and declarations for all other bidders, there exists a
critical value v c such that
∀v, v < v c ⇒ g j = ∅,
∀v, v > v c ⇒ g j = s,
We allow v c to be infinite if f (As,v)−1 (j) = ∅ for every v. Note that we do not
know whether j's bid is granted or not in case v = v c .

[Next verbatim source excerpt]

Our third property deals with a satisfied bidder: a satisfied bidder pays exactly
the critical value of Lemma 9.1, that is, the lowest value he could have declared
and still be allocated the goods he desires.
Critical

s ⊆ g j ⇒ p j = vc

Notice that Critical says, first, that the payment for a bid that is granted does not
depend on the amount of the bid, it depends only on the other bids. Then it says
that it is exactly equal to the critical value below which the bid would have lost.

Expanded PaperInterface specification

```
∀ {Bidder : Type u_1} {Item : Type u_2} [Fintype Bidder] [Fintype Item]
  (M : AppliedModelingLib.Auction.SingleMindedAcceptedMechanism Bidder Item),
  LOS02CombinatorialAuctions.SourceCritical.Feasible M →
    M.MonotonicityOn AppliedModelingLib.Auction.SingleMindedAcceptedMechanism.NonnegativeNonemptyProfile →
      ∀ (_C : M.NonnegativeCriticalValueWithInfinityCertificate)
        (D : Bidder → AppliedModelingLib.Auction.SingleMindedBid Item),
        AppliedModelingLib.Auction.SingleMindedAcceptedMechanism.NonnegativeNonemptyProfile D →
          ∀ i ∈ M.accepted D,
            ∀ (s : AppliedModelingLib.Auction.Bundle Item),
              Finset.Nonempty s →
                s ⊆ (D i).desired →
                  M.payment (Function.update D i { desired := s, value := (D i).value }) i ≤ M.payment D i
```

Lean proof endpoint (built separately)

LOS02CombinatorialAuctions.PaperInterface.lemma9_5

Recorded source-to-Spec screening **Verdict:** matches

Reason: Pinned Lemma 9.5 says that a granted ‘⟨s,v⟩‘ bid pays at least what the bidder would pay after declaring ‘⟨s’,v⟩‘ for ‘s’ ⊆ s‘, under Exactness, Monotonicity, and Critical. The target has those conditions, requires the original bid accepted and the approved nonempty subset domain, preserves its value, and states the equivalent inequality ‘payment(s’) ≤ payment(s)‘.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 10

Source locator: cited publication, lines 954-955

Verbatim source input

THEOREM 9.6. If a mechanism satisfies Exactness, Monotonicity, Participation
and Critical, then it is a truthful mechanism.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will
range over the bidders. Bidders are selfish, but rational, and trying to maximize their
utility in the final outcome. A bidder knows his own utility function (i.e., his type),
but this information is private and neither the auctioneer nor the other players have
access to it. The final result of an auction consists of two elements: an allocation
of the goods and a vector of payments from the bidders to the auctioneer, both of
which are functions of the bidders' declarations (i.e., bids). Formally, we have a
finite set G of k goods and an allocation is a partial function from G to P, that is, a
function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all
goods be allocated. Notice that the allocations produced by the Generalized Vickrey
Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always
total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions
from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types
for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice
that such a type allows for both complementarity and substitutability, but not for
externalities. Since the set 2i does not depend on i, we shall write 2. An element
of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is
also the set of messages that bidder i may send. A bidder may send any element of
2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t
to denote a (true) type, d to denote a message, T or D to denote vectors of n types
and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly
standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G
and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

We shall assume that bidders are single-minded and care only about one specific
(bidder-dependent) set of goods. If they do not get this set they value the outcome
at the lowest possible value 0. In other words, our bidders are restricted to one
single bid.
Definition 5.1. Bidder i is single-minded if and only if there is a set s ⊂ G of
goods and a value v ∈ R+ such that its type t can be described as: t(s 0) = v if s ⊆ s 0
and t(s 0) = 0 otherwise.
Note that single-minded bidders are not one parameter agents in the sense
of Archer and Tardos [2001], since they have the freedom of deciding which bundle
they bid for on top on the amount they are willing to pay. In Mu'alem and Nisan
[2002], truthfulness is shown to be attainable for allocation algorithms providing
better approximations if agents are assumed to be single-minded and one parameter. We shall denote by hs, vi the type just described. Note that a
↪ single-minded
bidder enjoys free disposal. We shall assume, in most of this article, that all bidders
are single-minded, that is, there are sets of goods si and nonnegative real numbers
v i such that bidder i is of type hsi , v i i. We shall denote by 6 the set of all singleminded types. The size of the set 6, contrary to the size of 2, is singly
↪ exponential
in k. A string of polynomial size will be enough to code the declarations of the
bidders: it will describe a set of goods and a value. In this setting, we identify bids

[Next verbatim source excerpt]

The general structure of the properties of interest is that they consider a given
set of single-minded types and vary one of those types. They restrict the changes
that can appear in the allocation or the payments as a result of such a change.
Let declarations be fixed, but arbitrary, for all bidders except j. Consider two
possible declarations for j: hs, vi and hs 0 , v 0 i. Given an allocation scheme f and a
payment scheme p, we shall consider the allocations and payments generated by
both declarations of j. Let gi be the set of goods obtained by bidder i if j declares
hs, vi, and gi0 the set he obtains if j declares hs 0 , v 0 i. Similarly denote by pi and pi0
the payments of i.
Our first property requires that the allocation, among single-minded bidders, be
exact, that is, a single-minded bidder either gets exactly the set of goods he desires,
nothing added, or he gets nothing. He never gets only part of what he requested. This
is a very natural property, when dealing with single-minded bidders: the valuation
of the bidder does not increase by giving him part of what he requested instead of
nothing or by giving him more than what he requested instead of just the bundle
he requested.
Exactness Either g j = s or g j = ∅.
In an exact allocation, we shall say that j's bid is granted in the first case, and denied
in the second case. In such a scheme, the allocation may be viewed as a set of bids
(or bidders) that is conflict-free, that is, the s coordinates have pairwise empty

[Next verbatim source excerpt]

Our next property, Monotonicity, also concerns only the allocation scheme. It
requires that, if j's bid is granted if he declares hs, vi, it is also granted if he
declares hs 0 , v 0 i for any s 0 ⊆ s, v 0 ≥ v. In other words, proposing more money for
fewer goods cannot cause a bidder to lose his bid. It follows that, similarly, offering
less money for more goods cannot cause a lost bid to win. Formally:
Monotonicity s ⊆ g j ,

s 0 ⊆ s,

v 0 ≥ v ⇒ s 0 ⊆ g 0j .

[Next verbatim source excerpt]

LEMMA 9.1. In a mechanism that satisfies Exactness and Monotonicity, given a bidder j, a set s of goods and declarations for all other bidders, there exists a critical value v c such that
∀v, v < v c ⇒ g j = ∅,
∀v, v > v c ⇒ g j = s,
We allow v c to be infinite if f (As,v)−1 (j) = ∅ for every v. Note that we do not know whether j's bid is granted or not in case v = v c .

[Next verbatim source excerpt]

Our third property deals with a satisfied bidder: a satisfied bidder pays exactly the critical value of Lemma 9.1, that is, the lowest value he could have declared and still be allocated the goods he desires.
Critical

s ⊆ g j ⇒ p j = vc

Notice that Critical says, first, that the payment for a bid that is granted does not depend on the amount of the bid, it depends only on the other bids. Then it says that it is exactly equal to the critical value below which the bid would have lost.

[Next verbatim source excerpt]

Our last property concerns the payment scheme. Together with Critical, it implies that the utility of no truthful bidder is negative. It concerns unsatisfied bidders, bids that are denied. We require that an unsatisfied bidder pay zero. The utility of an unsatisfied bidder is then zero. This is simply tuning the utility scales of the different bidders, or, ensuring that bidders may not lose by participating in the auction.
Participation

s 6⊆ g j ⇒ p j = 0

Expanded PaperInterface specification

```
∀ {Bidder : Type u_1} {Item : Type u_2} [Fintype Bidder] [Fintype Item]
  (M : AppliedModelingLib.Auction.SingleMindedAcceptedMechanism Bidder Item),
  LOS02CombinatorialAuctions.SourceCritical.Feasible M →
    M.MonotonicityOn AppliedModelingLib.Auction.SingleMindedAcceptedMechanism.NonnegativeNonemptyProfile →
      LOS02CombinatorialAuctions.SourceCritical.Participation M →
        ∀ (_C : M.NonnegativeCriticalValueWithInfinityCertificate),
          M.TruthfulOn AppliedModelingLib.Auction.SingleMindedAcceptedMechanism.NonnegativeNonemptyProfile
```

Lean proof endpoint (built separately)

LOS02CombinatorialAuctions.PaperInterface.theorem9_6

Recorded source-to-Spec screening **Verdict:** matches

Reason: Pinned Theorem 9.6 combines Exactness, Monotonicity, Participation, and Critical to obtain truthfulness. The target represents exactness by the accepted-set model plus 'Feasible', uses monotonicity and participation on the approved source request domain, includes the finite-or-infinite critical certificate, and concludes 'TruthfulOn' exactly on that legal nonnegative/nonempty single-minded domain.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 11

Source locator: cited publication, lines 993

Verbatim source input

THEOREM 10.2. The mechanism composed of the greedy allocation and payment schemes is truthful for single-minded bidders.

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will
range over the bidders. Bidders are selfish, but rational, and trying to maximize their
utility in the final outcome. A bidder knows his own utility function (i.e., his type),
but this information is private and neither the auctioneer nor the other players have
access to it. The final result of an auction consists of two elements: an allocation
of the goods and a vector of payments from the bidders to the auctioneer, both of
which are functions of the bidders' declarations (i.e., bids). Formally, we have a
finite set G of k goods and an allocation is a partial function from G to P, that is, a
function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all
goods be allocated. Notice that the allocations produced by the Generalized Vickrey
Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always
total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions
from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types
for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice
that such a type allows for both complementarity and substitutability, but not for
externalities. Since the set 2i does not depend on i, we shall write 2. An element
of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is
also the set of messages that bidder i may send. A bidder may send any element of
2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t
to denote a (true) type, d to denote a message, T or D to denote vectors of n types
and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly
standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G
and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

We shall assume that bidders are single-minded and care only about one specific
(bidder-dependent) set of goods. If they do not get this set they value the outcome
at the lowest possible value 0. In other words, our bidders are restricted to one
single bid.
Definition 5.1. Bidder i is single-minded if and only if there is a set s ⊂ G of
goods and a value v ∈ R+ such that its type t can be described as: t(s 0) = v if s ⊆ s 0
and t(s 0) = 0 otherwise.
Note that single-minded bidders are not one parameter agents in the sense
of Archer and Tardos [2001], since they have the freedom of deciding which bundle
they bid for on top on the amount they are willing to pay. In Mu'alem and Nisan
[2002], truthfulness is shown to be attainable for allocation algorithms providing
better approximations if agents are assumed to be single-minded and one parameter. We shall denote by hs, vi the type just described. Note that a
↪ single-minded
bidder enjoys free disposal. We shall assume, in most of this article, that all bidders
are single-minded, that is, there are sets of goods si and nonnegative real numbers
v i such that bidder i is of type hsi , v i i. We shall denote by 6 the set of all singleminded types. The size of the set 6, contrary to the size of 2, is singly
↪ exponential
in k. A string of polynomial size will be enough to code the declarations of the
bidders: it will describe a set of goods and a value. In this setting, we identify bids

[Next verbatim source excerpt]

A single-minded bidder declaring hs, ai, with s ⊆ G and a ∈ R+ will be said to
put out a bid b = hs, ai. We shall use s(b) and a(b) to denote the components of b
and call a(b) the amount of the bid b. As explained in Section 5, we identify bids
and bidders. Two bids b = hs, ai and b0 = hs 0 , a 0 i conflict if s ∩ s 0 6= ∅.
The algorithms we consider execute in two phases.
—In the first phase, the bids are sorted by some criterion. The algorithms of the
family are distinguished by the different criteria they use. Since there are n bids,
this phase takes time of the order of n log n. We assume a criterion, that is, a norm
is defined and the bids are sorted in decreasing order following this norm. Since
we shall have, in Theorem 10.2, to compare the sorted lists of bids of slightly
different auctions, we also assume a consistent treatment of ties, that is, bids
with equal norms. Formally, we shall assume that no two different bids have the
same norm, that is, there are no ties.
—In the second phase, a greedy algorithm generates an allocation. Let L be the
list of sorted bids obtained in the first phase. The first bid of L, say b = hs, ai is
granted, that is, the set s will be allocated to b and then the algorithm examines
each bid of L, in order, and grants it if it does not conflict with any of the bids
previously granted. If it does, it denies (i.e., does not grant) the bid. This phase
requires time linear in n.

[Next verbatim source excerpt]

a
.
Definition 7.1. The average amount per good of a bid b = hs, ai is |s|

[Next verbatim source excerpt]

We assume that the criterion used is average amount per good, the adaptation
to most other suitable greedy allocations is obvious. Informally, a bidder pays, per
good, the average price proposed by the first bid in the list L that is denied because
of this bid. Consider a bid j in L. Let c(j) be the average amount per good of j. We
shall denote by n(j) the first bid following j (bids are sorted in decreasing order,
that is, c(j) ≥ c(n(j))) that has been denied but would have been granted were it

not for the presence of j. Assume that such a bid exists. Notice that such a bid necessarily conflicts with j, and therefore:
n(j) = min{i | j < i, s(j) ∩ s(i) 6= ∅, ∀l, l < i, l 6= j, l granted ⇒ s(l) ∩ s(i) = ∅}.
Definition 10.1 Greedy Payment Scheme.
the first phase.

Let L be the sorted list obtained in

— j pays zero if his bid is denied or if there is no bid n(j),
—if there is an n(j) and j's bid hs, vi is granted, he pays |s| × c(n(j)).

[Next verbatim source excerpt]

should also increase the norm. We shall require that changing s to s 0 with s 0 ⊂ s or changing v to v 0 with v 0 > v increase the norm of a bid. Let us call this property bid-monotonicity. This is the only requirement we shall make. Many criteria satisfy it.

Expanded PaperInterface specification

∀ {Bidder : Type u_1} {Item : Type u_2} [inst : Fintype Bidder] [Fintype Item] [inst_2 : LinearOrder Bidder],
  LOS02CombinatorialAuctions.SourceGreedy.averageGreedyMechanism.TruthfulOn
    AppliedModelingLib.Auction.SingleMindedAcceptedMechanism.NonnegativeNonemptyProfile

Lean proof endpoint (built separately)

LOS02CombinatorialAuctions.PaperInterface.theorem10_2

Recorded source-to-Spec screening **Verdict:** matches

Reason: Pinned Theorem 10.2 makes the average-per-good greedy allocation with Definition 10.1's first-later-denied-bid payment truthful for single-minded bidders. The target uses that concrete average greedy mechanism and payment on the approved nonnegative/nonempty request domain, and asserts the same dominant-strategy utility inequality for every legal type and report. Its fixed 'LinearOrder' is the embedded user-approved consistent tie convention, not an unapproved added restriction.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 12

Source locator: cited publication, lines 1140-1141

Verbatim source input

12. No Payment Scheme Makes the Greedy Allocation a Truthful
Mechanism for Complex Bidders

[Next verbatim source excerpt]

Formally, we consider a set P of n bidders. The indices i, j, 1 ≤ i, j ≤ n, will
range over the bidders. Bidders are selfish, but rational, and trying to maximize their
utility in the final outcome. A bidder knows his own utility function (i.e., his type),
but this information is private and neither the auctioneer nor the other players have
access to it. The final result of an auction consists of two elements: an allocation
of the goods and a vector of payments from the bidders to the auctioneer, both of
which are functions of the bidders' declarations (i.e., bids). Formally, we have a
finite set G of k goods and an allocation is a partial function from G to P, that is, a
function a : G → P 0 , with P 0 = P ∪ {unallocated}, since we do not insist that all
goods be allocated. Notice that the allocations produced by the Generalized Vickrey
Auctions of Section 4 and by our Greedy algorithm of Section 7 are not always
total. The set of outcomes (i.e., allocations) is O = P 0G , the set of partial functions
from G to P. Since Gwe do not allow for externalities, the set 2i of the possible types
for bidder i is R+ 2 , where R+ is the set of all nonnegative real numbers. Notice
that such a type allows for both complementarity and substitutability, but not for
externalities. Since the set 2i does not depend on i, we shall write 2. An element
of 2 assigns a real nonnegative valuation to every possible bundle. The set 2 is
also the set of messages that bidder i may send. A bidder may send any element of
2, irrespective of his (true) type, that is, a bidder may lie. We shall typically use t
to denote a (true) type, d to denote a message, T or D to denote vectors of n types
and P for a payment vector, that is, a vector of n nonnegative numbers.
Since we assume the Independent Value Model and Quasi-Linear utilities, fairly
standard assumptions in the field, the utility, for a bidder of type t, of bundle s ⊆ G
and payment x is:
u = t(s) − x.

[Next verbatim source excerpt]

A single-minded bidder declaring hs, ai, with s ⊆ G and a ∈ R+ will be said to
put out a bid b = hs, ai. We shall use s(b) and a(b) to denote the components of b
and call a(b) the amount of the bid b. As explained in Section 5, we identify bids
and bidders. Two bids b = hs, ai and b0 = hs 0 , a 0 i conflict if s ∩ s 0 6= ∅.
The algorithms we consider execute in two phases.
—In the first phase, the bids are sorted by some criterion. The algorithms of the
family are distinguished by the different criteria they use. Since there are n bids,
this phase takes time of the order of n log n. We assume a criterion, that is, a norm
is defined and the bids are sorted in decreasing order following this norm. Since
we shall have, in Theorem 10.2, to compare the sorted lists of bids of slightly
different auctions, we also assume a consistent treatment of ties, that is, bids
with equal norms. Formally, we shall assume that no two different bids have the
same norm, that is, there are no ties.
—In the second phase, a greedy algorithm generates an allocation. Let L be the
list of sorted bids obtained in the first phase. The first bid of L, say b = hs, ai is
granted, that is, the set s will be allocated to b and then the algorithm examines
each bid of L, in order, and grants it if it does not conflict with any of the bids
previously granted. If it does, it denies (i.e., does not grant) the bid. This phase
requires time linear in n.

[Next verbatim source excerpt]

a
.
Definition 7.1. The average amount per good of a bid b = hs, ai is |s|

[Next verbatim source excerpt]

could consider any super-set of the single-minded types that grows only exponentially with k. A very natural idea is to consider players that send off
↪ single-minded
agent bidders to do their work. The agents play rationally, but individually, and
bring the goods and the payments due to the player. In the final analysis, a player
gets all the goods obtained by each of his agents and pays all the payments imposed
on each of his agents. A player's strategy is then a small (i.e., polynomial in k) set
of single-minded agents (i.e., bids).

[Next verbatim source excerpt]

Assume there are two goods a and b and two bidders Green and Red. Red is
single-minded and his type is h{a}, 10i. Red truthfully declares his type. Green is
interested in both b and the set {a, b}. His valuation is 0 for any allocation in which

[form-feed in source extraction]
598

D. LEHMANN ET AL.

he does not get b. It is v b for any allocation in which he gets b but not a, and it
is v ab if he gets both a and b, v ab > v b . Green's declaration is 0 for all allocations
that do not give him b, gb for all allocations that give him b but not a and gab for
the allocation in which he gets both a and b. Notice that four parameters describe
the auction: gab , gb , v ab and v b . Assume, furthermore, that 0 ≤ gb < 10. We reason
by contradiction and assume there is a payment scheme that makes truth-telling a
dominant strategy for Green. Let us consider two cases.

[Next verbatim source excerpt]

Definition 3.2. A mechanism h f, pi is truthful if and only if for every i ∈ P,
t ∈ 2 and any vector D of declarations, if D 0 is the vector obtained from D by
replacing the ith coordinate di by t, then: t(gi (D 0)) − pi (D 0) ≥ t(gi (D)) − pi (D).

Expanded PaperInterface specification

¬∃ (payment : LOS02CombinatorialAuctions.NegativeResults.GreenType → ℝ),
LOS02CombinatorialAuctions.NegativeResults.GreenTruthfulPayment payment

Lean proof endpoint (built separately)

LOS02CombinatorialAuctions.PaperInterface.section12

Recorded source-to-Spec screening **Verdict:** matches

Reason: The pinned Section 12 setup fixes Red's truthful '({a},10)' bid and studies Green's two coordinates '(v_ab,v_b)', with 'v_ab > v_b' and nonnegative reports satisfying '0 ≤ g_b < 10'. The target is exactly the source proof's fixed-Red two-good slice: an arbitrary report-based payment must make truthful reporting optimal for all stated true values and reports, and its negation gives the claimed impossibility. The approved fixed-priority tie convention supplies the comparison rule.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation