Comput Econ (2011) 38:173–204
DOI 10.1007/s10614-010-9228-9

Heuristics for Deciding Collectively Rational Consumption Behavior

Fabrice Talla Nobibon · Laurens Cherchye · Bram De Rock · Jeroen Sabbe · Frits C. R. Spieksma

Accepted: 26 May 2010 / Published online: 10 June 2010

Abstract We propose a directed graph for testing whether observed household consumption behavior satisfies the Collective Axiom of Revealed Preferences (CARP). More precisely, the data satisfy CARP if the graph allows a node-partitioning into two induced subgraphs that are acyclic. We prove that partitioning the obtained graph into two acyclic subgraphs is NP-complete. Next, we derive a necessary condition for CARP that can be verified in polynomial time, and we present an example to show that our necessary and sufficient conditions do not coincide. We also propose and implement fast heuristics for testing the sufficient condition. These heuristics can be used to check reasonably large data sets for CARP, and can be of particular interest

F. Talla Nobibon (✉) · F. C. R. Spieksma
Operations Research Group, University of Leuven, Naamsestraat 69, 3000 Leuven, Belgium
e-mail: Fabrice.TallaNobibon@econ.kuleuven.be

F. C. R. Spieksma
e-mail: Frits.Spieksma@econ.kuleuven.be

L. Cherchye
CentER, and University of Leuven, Campus Kortrijk and Center for Economic Studies, Tilburg University, E. Sabbelaan 53, 8500 Kortrijk, Belgium
e-mail: L.Cherchye@uvt.nl; laurens.cherchye@kuleuven-kortrijk.be

B. De Rock
ECARES and ECORE, Université Libre de Bruxelles, Avenue F. D. Roosevelt 50, CP 114, 1050 Brussels, Belgium
e-mail: bderock@ulb.ac.be

J. Sabbe
Campus Kortrijk and Center for Economic Studies, University of Leuven, E. Sabbelaan 53, 8500 Kortrijk, Belgium
e-mail: jeroen.sabbe@kuleuven-kortrijk.be

when used prior to computationally demanding approaches. Finally, based on computational results for both real-life and synthetic data, we conclude that our heuristics are effective in testing CARP.

Keywords Collective axiom of revealed preference · Pareto efficiency · Directed graph · Graph coloring · Heuristics · NP-complete

1 Introduction

The economics literature has paid notable attention to modeling household consumption behavior. In this respect, Chiappori's (1988, 1992) collective model of household consumption has become increasingly popular in recent years. The model explicitly recognizes that a household consists of multiple individuals (household members and/or decision makers) with their own rational preferences. In this sense, this collective approach contrasts with the conventional unitary approach, which models households as if they were single decision makers.

The distinguishing feature of the collective model is that it only assumes Pareto efficiency of the collective household decisions, i.e. the intra-household allocation process yields consumption outcomes such that no household member can be made better off without making another member worse off. The use of Pareto efficiency as the sole assumption is in sharp contrast with usual cooperative models of household consumption behavior, which typically combine multiple bargaining assumptions (see Lundberg and Pollak (2007) for a recent survey).

In the following, we concentrate on a general collective consumption model, which accounts for consumption externalities and public consumption within the household (see Browning and Chiappori (1998); Donni (2008) provides a neat overview of alternative collective consumption models). In the present context, public consumption of a certain good, which must be distinguished from private consumption, means that consumption of this good by one household member does not affect the supply available for another household member, and no individual can be excluded from consuming it (at least if one wants to maintain the household). Of course, some commodities may be partly publicly consumed (e.g. car use for a family trip) and partly privately consumed (e.g. car use for work). Next, consumption externalities refer to the fact that one household member gets utility from another member's consumption (e.g. the wife enjoys her husband's nice clothes).

The general collective model provides a useful starting point for testing Pareto efficiency of household collective consumption decisions: a rejection of the corresponding empirical restrictions can be interpreted as a rejection of the efficiency assumption. Moreover, given that all cooperative models use Pareto efficiency as a basic assumption and since it is also a natural benchmark in most non-cooperative settings, this test can also serve as basic input for these models.

Cherchye et al. (2007, 2010) propose a testable nonparametric revealed preference condition for data consistency with the collective consumption model.[1] This condition (see Sect. 2) is known as the Collective Axiom of Revealed Preferences (CARP). CARP is a necessary and sufficient condition, i.e., observed household consumption behavior is consistent with the collective consumption model if and only if observed household consumption behavior satisfies CARP Cherchye et al. (2010). Because it uses minimal prior structure, checking CARP consistency implies a "pure" test of Pareto efficiency. Such a test can provide a most convincing case for the goodness of, in general, the Pareto efficiency assumption and, in particular, the collective consumption model.

Essentially, CARP provides the collective counterpart of the unitary concept GARP (Generalized Axiom of Revealed Preferences; see Sect. 2). The issue of testing data consistency with GARP has attracted considerable attention in the literature on the unitary model of household consumption; see Varian (2006) and Cherchye et al. (2009) for recent surveys. This paper complements this rich literature by focusing on testing CARP.

More specifically, we consider the computational issues involved in checking CARP and we propose a graph-theoretical representation of the CARP conditions. Following this approach, we derive a sufficient condition for CARP consistency, of which verification is shown to be an NP-complete problem, and a necessary condition for CARP consistency, which can be verified in polynomial time. Moreover, our graph-theoretical approach allows us to propose and implement heuristics that quickly test our sufficient condition for CARP. A consequence of attempting to test CARP quickly, is that the outcome of a heuristic may be inconclusive, i.e., it is possible that after running the heuristic it is still not clear whether the data satisfy CARP. By performing computational experiments, however, we show that a vast majority of real-life instances is susceptible to our approach. This leads us to conclude that heuristics are relevant for testing CARP, particularly for large data sets; see Cherchye et al. (2008) and Deb (2008b) for recent discussions of the relevance of testing CARP for large instances.

Our graph-theoretical approach complements recent work of Cherchye et al. (2008), who formulate the computational problem of verifying CARP as an Integer Programming (IP) problem. These authors show practical usefulness of this IP test for empirically evaluating the collective model. Using the CPLEX IP solver, they perform their test on real-life data sets that are of reasonably large size when compared to existing nonparametric studies. It is well-known, however, that solving IP problems with exact implicit enumeration methods is computationally demanding. The approach presented in this study is particularly useful for (large) instances where Cherchye et al.'s IP approach takes a long time, or is unable to decide whether or not consumption behavior is collectively rational (see Sect. 5). In fact, even if the proposed heuristics

[1] Throughout, nonparametric analysis stands for revealed preference analysis in the tradition of, among others, Afriat (1967); Diewert (1973) and Varian (1982). Essentially, nonparametric (revealed preference) conditions allow for testing data consistency with a particular consumption model starting from a finite set of consumption observations, while avoiding the use of a (parametric) functional structure for the consumption decision process.

fail to directly decide CARP, their outputs can be used to reduce the size of the IP to be solved, by fixing some variables.

In another study, Deb (2008b) proposes a different heuristic for testing the collective model. This heuristic pertains to a condition for collective rationality which he shows to be NP-complete. Yet, Deb's collective rationality condition is sufficient but not necessary for CARP; that is, data satisfying this condition satisfy CARP, but not necessarily vice versa. We will show that our sufficient condition extends the condition proposed by Deb. Specifically, all data sets that pass Deb's test also pass our test, but the opposite conclusion does not hold.

At a more general level, our analysis demonstrates the usefulness of operations research techniques to implement nonparametric (revealed preference) conditions for economic decision behavior. In fact, our insights on testing CARP consistency can also be instrumental for designing operational tests in alternative settings. For instance, they readily extend to the general case of multi-person group consumption. See Chiappori and Ekeland (2006, 2009) for discussion on the relevance of the collective model within the context of group consumption. To ease our exposition, the theoretical discussion in the following sections focuses on two-person households. Generalizations for M-member groups ($M \geq 2$) are fairly easy and can be obtained along the lines of Cherchye et al. (2007, supplemental material). The sufficient condition as well as the heuristics derived in this paper can easily be extended to deal with the general case of M-member groups. Next, the nonparametric approach for analyzing collective consumption behavior is closely related to the literature on testable nonparametric conditions of general equilibrium models, which deals with formally similar issues. See, for example, Brown and Matzkin (1996); Brown and Shannon (2000) and, for more recent developments, Carvajal et al. (2004). Lastly, our results for the collective consumption model can also be relevant for nonparametric production analysis. See Cherchye et al. (2008), who adopt a formally similar collective model for analyzing economies of scope in the context of multi-output production.

The rest of the paper unfolds as follows. Sect. 2 defines collective rationality, and the corresponding CARP condition. Section 3 introduces the graph formulation, establishes a sufficient and a necessary condition for CARP. It also discusses the complexity of testing these conditions. Section 4 presents the heuristics. Section 5 discusses the computational results. Section 6 concludes.

2 Rationality Conditions

Household consumption behavior that is consistent with the collective consumption model is said to be collectively rational. The empirical condition for collective rationality requires that a collective rationalization is possible, i.e. the data can be made consistent with the collective consumption model. As indicated above, a collective rationalization of the data is possible if and only if the data are consistent with CARP. This section provides formal definitions of the different concepts of rationalization. We first present a preliminary discussion of unitary rationality and the corresponding Generalized Axiom of Revealed Preference (GARP). This will set the stage for our subsequent discussion of the collective model.

2.1 Unitary Rationality and GARP

Suppose we observe T individual choices of N-valued bundles. For each observation t the vector $q_t \in \mathbb{R}_+^N$ (with non-negative components) records the chosen quantities under the prices $p_t \in \mathbb{R}_{++}^N$ (with strictly positive components). We let $S = \{(p_t, q_t); t \in \mathbb{T} \equiv \{1 \ldots, T\}\}$ be the corresponding set of T observations, also referred to as the data. For ease of exposition, the scalar product $p_t' q_t$ is written as $p_t q_t$.

We recall that the unitary model assumes that the household behaves as a single decision maker, i.e. the observed household quantities maximize a well-behaved household utility function U subject to the household budget constraint.[2] We obtain the following condition for household behavior to be consistent with the unitary model.

Definition 1 (unitary rationalization) Let $S = \{(p_t, q_t); t \in \mathbb{T}\}$ be a set of observations. A utility function U provides a unitary rationalization of S if for each observation t we have $U(q_t) \geq U(z)$ for all quantities z with $p_t q_t \geq p_t z$.

Varian (1982) (based on Afriat (1967)) demonstrates that a data rationalizing utility function exists if and only if the observed set S is consistent with the Generalized Axiom of Revealed Preference (GARP). Essentially, GARP imposes empirical restrictions on revealed preference relations R_0 and R. First, $q_s R_0 q_t$ means that the decision maker (directly) reveals her or his preference for the quantities q_s over the quantities q_t. Next, $q_s R q_t$ represents the transitive closure, that is $q_s R q_t$ means that there exists a (possibly empty) sequence $u, \ldots, w \in T$ with $q_s R_0 q_u, q_u R_0 q_v, \ldots$ and $q_w R_0 q_t$. We can now define the GARP condition that applies to the unitary model.

Definition 2 (GARP) Let $S = \{(p_t, q_t); t \in \mathbb{T}\}$ be a set of observations. The set S satisfies GARP if there exist relations R_0, R that meet:

Rule 1: For $s, t \in \mathbb{T}$: if $p_s q_s \geq p_s q_t$, then $q_s R_0 q_t$;
Rule 2: For $s, t \in \mathbb{T}$: if $p_t q_t > p_t q_s$, then $\neg(q_s R q_t)$.

Testing GARP proceeds in two steps. First, one recovers the relations R_0 and (the transitive closure) R on the basis of Rule 1. Subsequently, one checks Rule 2, which requires $p_t q_t \leq p_t q_s$ for $q_s R q_t$. Varian (1982, p. 949) presents an efficient testing algorithm. Dobell (1965); Varian (1982) and Chung-Piaw and Vohra (2003) discus graph-theoretical representations of GARP. Our graph-theoretical formulation extends these studies by considering formally similar representations for the collective consumption model.

2.2 Collective Rationality

Once again, we consider a household that purchases the (non-zero) N-vector of quantities $q \in \mathbb{R}_+^N$ with corresponding prices $p \in \mathbb{R}_{++}^N$, and we start from a set of observations S. As indicated in the introduction, we focus on a two-member household to keep

[2] In the context of the unitary model, well-behavedness means that the utility functions are locally non-satiated; see, for example, Varian (1982). In the context of the collective model, it means that utility functions satisfy local collective non-satiation; this is the collective consumption analogue of the standard local non-satiation concept for the unitary model. See Cherchye et al. (2010) for more discussion.

our exposition simple. All goods can be consumed privately (e.g. member 1 uses the car alone), publicly (e.g. member 1 and member 2 use the car together), or both. Generally, we assume that the empirical analyst has no information on the decomposition of the observed q. That is, we need to consider all allocations $q = q^1 + q^2 + q^h$ for q the (observed) aggregate quantities, q^1 and q^2 the (unobserved) private quantities of each household member, and q^h the (unobserved) public quantities.

The collective model explicitly recognizes the individual preferences of the household members. Because we account for consumption externalities, these preferences may depend not only on the own private and public quantities, but also on the other individual's private quantities. Formally, this means that the preferences of each household member m ($m = 1, 2$) can be represented by a well-behaved utility function of the form U^m that is defined in the arguments q^1, q^2 and q^h. Note that we do not demand that these utility functions are concave.[3]

For aggregate quantities q, we define feasible personalized quantities $\widehat{q}$ as

$$\widehat{q} = \left(\mathsf{q}^1, \mathsf{q}^2, \mathsf{q}^h\right) \text{ with } \mathsf{q}^1, \mathsf{q}^2, \mathsf{q}^h \in \mathbb{R}^n_+ \text{ and } \mathsf{q}^1 + \mathsf{q}^2 + \mathsf{q}^h = q.$$

Each $\widehat{q}$ captures a feasible decomposition of the aggregate quantities q into private quantities (q^1 and q^2) and public quantities (q^h). This reflects that our model allows for general preferences that depend on private and public consumption. In the following, we consider feasible personalized quantities because we assume the minimalistic prior that only the aggregate quantity bundle q and not the "true" personalized quantities are observed. Throughout, we will use that each $\widehat{q}$ defines a unique q.

Given this, a collective rationalization of S requires the existence of utility functions U^1 and U^2 such that each observed consumption bundle can be characterized as Pareto efficient. Thus, we get the following definition, which has an analogous structure as Definition 1.

Definition 3 (collective rationalization) Let $S = \{(p_t, q_t); t \in \mathbb{T}\}$ be a set of observations. A pair of utility functions U^1 and U^2 provides a collective rationalization of S if for each observation t there exist feasible personalized quantities $\widehat{q_t}$ such that $U^m(\widehat{z}) > U^m(\widehat{q_t})$ implies $U^l(\widehat{z}) < U^l(\widehat{q_t})$ $(m \neq l)$ for all feasible personalized quantities $\widehat{z}$ with $p_t q_t \geq p_t z$.

2.3 Collective Axiom of Revealed Preference (CARP)

This section presents CARP, which provides a testable nonparametric necessary and sufficient condition for a collective rationalization of the data as described in the previous section. We refer to Cherchye et al. (2007, 2010) for detailed discussions on CARP and the equivalent results.

Essentially, CARP imposes empirical restrictions on hypothetical member-specific preference relations H_0^m and H^m, which have a similar interpretation as the revealed

[3] Indeed, it has been argued that in the presence of externalities (i.e. the utility of one member depends on the private consumption of the other member) this assumption of concave utility functions (or, alternatively, convex preferences) is problematic. See, for example, Starr (1969) and Starret (1972).

preference relations R_0 and R for the unitary model. In this case, the hypothetical relations H_0^m and H^m represent feasible specifications of the true individual preference relations that are consistent with the information revealed by the set of observations S. First, $q_s H_0^m q_t$ means that we "hypothesize" that member m (directly) prefers the quantities q_s over the quantities $q_t, m = 1, 2$. Next, $q_s H^m q_t$ represents the transitive closure, that is $q_s H^m q_t$ means that there exists a (possibly empty) sequence $u, \ldots, w \in T$ with $q_s H_0^m q_u, q_u H_0^m q_v, \ldots$ and $q_w H_0^m q_t$. Notice that, while the "true" preferences are -of course- expressed in terms of the feasible personalized quantities $\widehat{q}$ (i.e. member m prefers q_s over q_t only if $U^m(\widehat{q_s}) \geq U^m(\widehat{q_t})$), the hypothetical preferences only use observable information (captured by the observed aggregate prices p and quantities q in the set S). This naturally complies with the assumption that in the general model we have no information concerning the feasible personalized quantities.

Given this notion of hypothetical preference relations, we can define CARP. The next definition, which reformulates Definition 6 of Cherchye et al. (2010), gives us a condition that can be empirically tested on aggregate price-quantity information. Moreover, these authors show that there exists a collective rationalization of the data in terms of Definition 3 if and only if the data is consistent with CARP. As such, we obtain the desired test of Pareto efficiency.

Definition 4 (CARP) Let $S = \{(p_t, q_t); t \in \mathbb{T}\}$ be a set of observations. S satisfies CARP if there exist hypothetical relations H_0^m, H^m for each member $m \in \{1, 2\}$ that meet:

Rule 1: For $s, t \in \mathbb{T}$, if $p_s q_s \geq p_s q_t$ then either $q_s H_0^1 q_t$ or $q_s H_0^2 q_t$;

Rule 2: $\begin{cases} \text{(a) For } s, t \in \mathbb{T}, \quad \text{if } p_s q_s \geq p_s q_t \text{ and } q_t H^m q_s \text{ then } q_s H_0^l q_t \text{ with } l \neq m, \\ \text{(b) For } s, t_1, t_2 \in \mathbb{T}, \quad \text{if } p_s q_s \geq p_s (q_{t_1} + q_{t_2}) \text{ and } q_{t_1} H^m q_s; \\ \quad \text{then } q_s H_0^l q_{t_2} \text{ with } l \neq m \end{cases}$

Rule 3: $\begin{cases} \text{(a) For } s, t \in \mathbb{T}, \quad \text{if } p_s q_s > p_s q_t \text{ then either } \neg(q_t H^1 q_s) \text{ or } \neg(q_t H^2 q_s) \\ \text{(b) For } s, t_1, t_2 \in \mathbb{T}, \quad \text{if } p_s q_s > p_s (q_{t_1} + q_{t_2}) \text{ then either } \neg(q_{t_1} H^1 q_s) \\ \quad \text{or } \neg(q_{t_2} H^2 q_s). \end{cases}$

This CARP condition has a similar structure as the unitary GARP condition in Definition 2. Specifically, GARP states (in casu unitary) rationality conditions in terms of the preference information that is revealed by the observed price and quantity data. CARP does the same, but now the revealed preference information is understood in terms of the collective model of household consumption and, thus, pertains to the individual household members. It can be verified that a data set S satisfies CARP if it satisfies GARP, but not vice versa. In other words, CARP consistency is necessary but not sufficient for GARP consistency.

Interestingly, CARP has a direct interpretation in terms of the Pareto efficiency requirement that underlies collective rationality. Rule 1 states that, if the quantities q_s were chosen while the quantities q_t were equally attainable (under the prices p_s), then it must be that at least one member prefers the quantities q_s over the quantities q_t (i.e. $q_s H_0^1 q_t$ or $q_s H_0^2 q_t$). Rule 2a can also be interpreted in terms of Pareto efficiency as follows: if member m prefers q_t over q_s for the bundle q_t not more expensive than q_s (i.e. $p_s q_s \geq p_s q_t$), then the choice of q_s can be rationalized only if the other member, member l, prefers q_s over q_t. Indeed, if this last condition were not satisfied, then the bundle q_t (under the given prices p_s and outlay $p_s q_s$) would imply a Pareto improvement over the chosen bundle q_s. Analogously, Rule 2b states that, if the summed bundle $q_{t_1} + q_{t_2}$ is attainable and member m prefers q_{t_1} over q_s, then Pareto efficiency requires that the other member (member l) prefers q_s over q_{t_2}. Finally, Rule 3 complements Rule 2 and its interpretation in terms of Pareto efficiency is the following. Rule 3a states that, if q_t was attainable when q_s was chosen, then it cannot be that both members prefer q_t over q_s; otherwise Pareto improvements would have been possible (under the given prices p_s and outlay $p_s q_s$), which conflicts with collective rationality. Similarly, Rule 3b states that, if $q_{t_1} + q_{t_2}$ was attainable when q_s was chosen, then it cannot be that member m prefers q_{t_1} over q_s while, at the same time, member l prefers q_{t_2} over q_s. In the rest of this paper, we will refer to the inequality $p_s q_s \geq p_s(q_{t_1} + q_{t_2})$ as double-sum inequality. The following example, which we also use in the next section, illustrates CARP.

Example 1 Consider a situation with three goods ($N = 3$) and two household members ($M = 2$), with the following three observed price-quantity combinations ($T = 3$):

1: $q_1 = (8 \quad 2 \quad 2)'$ and $p_1 = (6 \quad 2 \quad 2)'$
2: $q_2 = (1 \quad 8 \quad 3)'$ and $p_2 = (2 \quad 6 \quad 1)'$
3: $q_3 = (1 \quad 2 \quad 8)'$ and $p_3 = (2 \quad 3 \quad 5)'$

Observe that the following inequalities are the only inequalities implied by this data set.

I_1: $p_s q_s \geq p_s q_t$ for each pair $s, t \in \{1, 2, 3\}$
I_2: $p_1 q_1 > p_1(q_2 + q_3)$
I_3: $p_2 q_2 > p_2(q_1 + q_3)$

Consider H_0^1 and H_0^2 defined as follows. For each observation $s \in \{1, 2, 3\}$, we have $q_s H_0^m q_s$ for $m = 1, 2$. Moreover, $q_1 H_0^1 q_2, q_1 H_0^1 q_3$ and $q_3 H_0^1 q_2$ while $q_2 H_0^2 q_1, q_3 H_0^2 q_1$ and $q_2 H_0^2 q_3$. One can verify that this specification of H_0^1 and H_0^2 satisfies Rule 1–3 in Definition 4. As such, we conclude that the data satisfies CARP.

One concluding remark pertains to the fact that CARP only uses information on the (observed) aggregate quantities q, and not on the (unobserved) private quantities q^1 and q^2 and (unobserved) public quantities q^h. Because CARP provides a necessary and sufficient condition for a collective rationalization of the data, this actually means that the distinction between private consumption (with or without externalities) and public consumption is irrelevant in view of empirical tests of the collective consumption model. See also Cherchye et al. (2010) for a detailed treatment of this issue.

3 A Graph-Theoretic Formulation

In this section, we show how to build from the data S a directed graph $G(S) = (V(S), A(S))$ and use it to derive the following sufficient condition for CARP. If the nodes of $V(S)$ can be partitioned into two subsets such that each induced subgraph is acyclic, then the data satisfy CARP. By *induced subgraph*, we mean a subset of nodes of G together with any arcs whose both endpoints are in that subset; an *acyclic subgraph* is a subgraph which does not contain a cycle. Subsequently, we present a necessary condition for CARP. We also provide an example which shows that there exist instances of data S for which neither the sufficient condition nor the necessary condition presented here are satisfied, while there exist hypothetical relations H_0^1, H_0^2 satisfying Rule 1–3 in Definition 4. Finally, we prove that while checking the necessary condition can be achieved in polynomial-time, deciding whether a partition of nodes into two acyclic subgraphs exists for the graph $G(S)$ is NP-complete. For reasons of notational convenience, we will simply write G, V, and A instead of $G(S)$, $V(S)$, and $A(S)$ respectively.

3.1 Construction of the Graph

Given a set of observations $S = \{(p_t, q_t); t \in \mathbb{T}\}$, each pair of distinct observations (s, t) with $s, t \in \mathbb{T}$ represents a node in V if $p_s q_s \geq p_s q_t$. Hence, the nodes (s, t) and (t, s) (if they exist) are different. No other nodes exist in V. A node (s, t) is said to be involved in a double-sum inequality if there exists a node $(s, u) \in V$ such that $p_s q_s \geq p_s(q_t + q_u)$. Such nodes (s, t) and (s, u) will be called *double-sum nodes*. Essentially, the nodes in V bear the following relations to Rules 1–3 in the earlier Definition 2 of CARP. Any node $(s, t) \in V$ (with $p_s q_s \geq p_s q_t$) is relevant for Rules 1, 2a and 3a. Similarly, any pair of double-sum nodes (s, t) and (s, u) (with $p_s q_s \geq p_s(q_t + q_u)$) is relevant for Rules 2b and 3b.

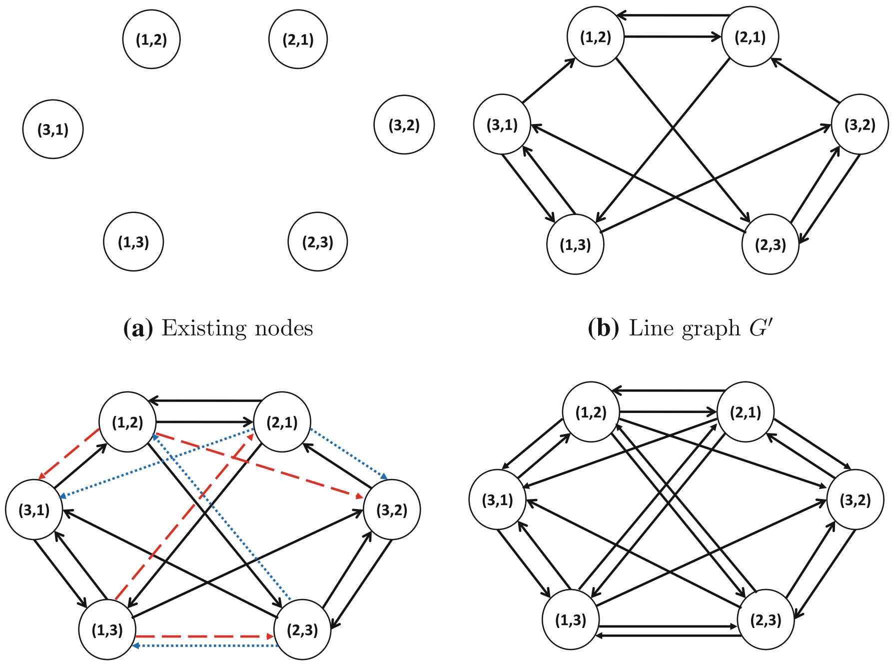
Given all this, the set of arcs A is defined in two stages:

a: First of all, we draw an arc from a node (s, t) to a node (u, v) whenever $t = u$. The resulting graph is denoted by $G' = (V, A')$ and is a *line graph* Gross and Yellen (2004).

b: Second, for any given three distinct observations $s, t_1, t_2 \in \mathbb{T}$, verify whether $p_s q_s \geq p_s(q_{t_1} + q_{t_2})$ and whether there exist two observations $u, v \in \mathbb{T}$ (respectively $u', v' \in \mathbb{T}$) such that $(t_1, u), (v, s) \in V$ (respectively $(t_2, u'), (v', s) \in V$). If so, we distinguish two different cases:
   - $(t_1, u) \neq (v, s)$ (respectively $(t_2, u') \neq (v', s)$)
     – If there is a path in G' from (t_1, u) to (v, s) (respectively from (t_2, u') to (v', s)), then we draw an arc from (s, t_2) to (t_1, u) (respectively from (s, t_1) to (t_2, u')). Notice that the nodes (s, t_1) and (s, t_2) exist in V.
   - $(t_1, u) = (v, s)$ (respectively $(t_2, u') = (v', s)$)
     – We draw an arc from (s, t_2) to (t_1, u) (respectively from (s, t_1) to (t_2, u')).

The directed graph $G = (V, A)$ is then defined by the set of nodes V described above and the set of arcs A described by a) and b). The arcs defined in b) will be called "double-sum arcs". Notice that if there is no extra arc defined in b), then $G = G'$. Observe that in the construction of G, we associate a node to a pair of observations. This allows us to take into account relationships between three observations as formulated in Rule 2 and Rule 3. The following example illustrates the above construction.

Example 2 We consider the data of Example 1. The first set of inequalities (I_1) implies the existence of all possible nodes in the graph. Figure 1a represents the nodes of



(a) Existing nodes **(b)** Line graph G'

(c) double-sum arcs added **(d)** Final graph G

Fig. 1 Illustration of the construction of G. **a** Existing nodes; **b** Line graph G'; **c** double-sum arcs added; **d** Final graph G

graph G. The first step in the construction of the set of arcs leads to the line graph given by Fig. 1b. Next, the double-sum arcs are added to the line graph. In Fig. 1c, the dashed arcs — — —▷ correspond with the double-sum inequality I_2 (i.e. $p_1q_1 > p_1(q_2+q_3)$) while the dashed arcs $\cdots\cdots$▷ correspond with I_3 (i.e. $p_1q_1 > p_1(q_2+q_3)$). Finally the graph G is depicted in Fig. 1d.

Notice that the construction of G is done in polynomial time. In fact, an algorithm that finds the set V of nodes and determines double-sum nodes is implemented to run in time $O(T^3)$. Having the set of nodes, the line graph is immediate. To build the double-sum arcs, we proceed as follows. For a given node (s, t) involved in a double-sum inequality $p_sq_s \geq p_s(q_t + q_u)$, we use Dijkstra's algorithm Ahuja et al. (1993) to find all the nodes which are such that there is a path in G' from (s, t) to those nodes. Among those nodes, we identify the nodes ending with s (these are nodes $(., s)$) and draw an arc from (s, u) to the node $(t, .)$ appearing in each path.

3.2 A Sufficient Condition for CARP

In this section, we show that if the graph G can be node-partitioned into two acyclic subgraphs, then the set S of observations satisfies CARP; that is there exist H_0^1

and H_0^2 satisfying Rule 1–3 in Definition 4. In other words, when we can color each node of the directed graph G with one of the two colors red or blue, such that $V = V_B \cup V_R$, $V_B \cap V_R = \emptyset$, where V_B (respectively V_R) is the set of nodes colored blue (respectively red); and the induced subgraphs $G_B = (V_B, A_B)$, $G_R = (V_R, A_R)$ are each acyclic, then there exists hypothetical relations H_0^1 and H_0^2 that satisfy Rule 1–3 in Definition 4.

An equivalent way of phrasing this sufficient condition is as follows: can we color each node of G red or blue such that no monochromatic cycle exists? (A monochromatic cycle is a cycle containing nodes of the same color). For an arbitrary directed graph G, the problem of node-partitioning the graph into two acyclic induced subgraphs is proven to be NP-complete by Deb (2008a). Results for undirected graphs can be found in Chen (2000) (who gives an efficient algorithm to minimize the number of acyclic subgraphs), and more recently by Chang et al. (2004) (who study the complexity of the problem for specific graph classes).

Theorem 1 *If the directed graph G can be node-partitioned into two acyclic subgraphs then the set S of observations satisfies CARP.*

Proof Suppose that G can be partitioned into two acyclic subgraphs $G_B = (V_B, A_B)$ and $G_R = (V_R, A_R)$. From this partition we infer H_0^1 and H_0^2 as follows.

H_0^1: (i) $q_s H_0^1 q_t$ if and only if $(s, t) \in V_B$ and (ii) $q_s H_0^1 q_s$ for all $s \in \mathbb{T}$.

H_0^2: (i) $q_s H_0^2 q_t$ if and only if $(s, t) \in V_R$ and (ii) $q_s H_0^2 q_s$ for all $s \in \mathbb{T}$.

In other words, for each observation $s \in \mathbb{T}$, $q_s H_0^1 q_s$ and for each node (s, t) which is colored blue, we have $q_s H_0^1 q_t$. For each node (s, t) which is colored red, we have $q_s H_0^2 q_t$ and for each observation $s \in \mathbb{T}$, $q_s H_0^2 q_s$. We are now going to check that Rule 1–3 hold.

Rule 1: Let $s, t \in \mathbb{T}$ be two distinct observations such that $p_s q_s \geq p_s q_t$. Then $(s, t) \in V = V_B \cup V_R$, which implies that $(s, t) \in V_B$ or $(s, t) \in V_R$, and hence $q_s H_0^1 q_t$ or $q_s H_0^2 q_t$ by construction of H_0^1 and H_0^2. Moreover, for each observation $s \in \mathbb{T} q_s H_0^i q_s (i = 1, 2)$ by definition. Thus Rule 1 is satisfied.

Rule 2**a**: Clearly, this rule is satisfied for a single observation s. Let $s, t \in \mathbb{T}$ be two distinct observations such that $p_s q_s \geq p_s q_t$ and $q_t H^1 q_s$. Now, $p_s q_s \geq p_s q_t$ implies that $(s, t) \in V$ and $q_t H^1 q_s$ implies that there exist observations $u, u_0, u_1, \ldots, u_k, v \in \mathbb{T}$ such that (t, u), (u, u_0), $(u_0, u_1), \ldots, (u_{k-1}, u_k)$, (u_k, v), $(v, s) \in V$. By construction of G, there is a cycle containing the nodes (s, t), (t, u), (u, u_0), $(u_0, u_1), \ldots, (u_{k-1}, u_k)$, (u_k, v) and (v, s). Since $q_t H^1 q_s$, all the nodes (t, u), (u, u_0), $(u_0, u_1), \ldots, (u_{k-1}, u_k)$, (u_k, v), (v, s) are in V_B. $G_B = (V_B, A_B)$ is an acyclic subgraph implies that $(s, t) \in V_R$ and hence $q_s H_0^2 q_t$. Notice that a similar reasoning is applied to show that if $p_s q_s \geq p_s q_t$ and $q_t H^2 q_s$ then $q_s H_0^1 q_t$ for any distinct observations s and t. This completes the proof that **Rule** 2**a** is satisfied.

Rule 2**b**: Suppose $s, t_1, t_2 \in \mathbb{T}$. Notice that if $s = t_1$ or $s = t_2$ then $p_s q_s < p_s(q_{t_1} + q_{t_2})$ and if $t_1 = t_2$ checking this rule becomes equivalent to checking **Rule** 2**a**. Hence, we assume that s, t_1, t_2 are three distinct observations such that $p_s q_s \geq p_s(q_{t_1} + q_{t_2})$ and $q_{t_1} H^1 q_s$. Now, $p_s q_s \geq p_s(q_{t_1} + q_{t_2})$

implies that (s, t_1) and (s, t_2) belong to V. Also, $q_{t_1} H^1 q_s$ implies that there exists $u, v \in \mathbb{T}$ such that (t_1, u), $(v, s) \in V$ and either $(t_1, u) \neq (v, s)$ and there is a path from (t_1, u) to (v, s) or $(t_1, u) = (v, s)$. By construction of G, there is a cycle containing the node (s, t_2) and (t_1, u). Remark that if $(t_1, u) = (v, s)$ then that cycle contains only two nodes which are (t_1, s) and (s, t_2). Moreover, $q_{t_1} H^1 q_s$ indicates that all the nodes of the path from (t_1, u) to (v, s) (included) are in V_B or $(t_1, s) \in V_B$ if $(t_1, u) = (v, s)$. Since $G_B = (V_B, A_B)$ is an acyclic subgraph, $(s, t_2) \in V_R$ and $q_s H_0^2 q_{t_2}$. As in the proof of **Rule** 2**a**, the symmetry between H_0^1 and H_0^2 allows us to conclude that if $p_s q_s \geq p_s(q_{t_1} + q_{t_2})$ and $q_{t_1} H^2 q_s, q_s H_0^1 q_{t_2}$ for any three distinct observations s, t_1, t_2. This completes the proof of **Rule** 2**b**.

Rule 3**a**: Since $V_B \cap V_R = \emptyset$ and $p_s q_s = p_s q_s$ for each $s \in \mathbb{T}$, this property holds.

Rule 3**b**: Suppose that $s, t_1, t_2 \in \mathbb{T}$ are three distinct observations such that $p_s q_s > p_s(q_{t_1} + q_{t_2})$ and $q_{t_1} H^1 q_s$ and $q_{t_2} H^2 q_s$. Now, $p_s q_s > p_s(q_{t_1} + q_{t_2})$ implies that $(s, t_1) \in V = V_B \cup V_R$. From $q_{t_2} H^2 q_s$ and **Rule** 2**b**, we know that $(s, t_1) \in V_B$. $q_{t_1} H^1 q_s$ implies that there exists $u, v \in \mathbb{T}$ such that (t_1, u), $(v, s) \in V$ and either $(t_1, u) \neq (v, s)$ and there is a path from (t_1, u) to (v, s) in $G_B = (V_B, A_B)$ or $(t_1, u) = (v, s)$ and $(t_1, s) \in V_B$. $(s, t_1) \in V_B$ implies that $G_B = (V_B, A_B)$ contains a cycle. This contradicts the fact that G_B is acyclic.

We have shown that if the graph G can be partitioned into two acyclic subgraphs, then from these subgraphs, we can infer H_0^1 and H_0^2 satisfy Rules 1–3. □

The next example illustrates our sufficient condition.

Example 3 Consider the graph G of Fig. 1d built from data of Example 1. A possible coloring of nodes into two acyclic subgraphs is to color the nodes (1,2), (1,3) and (3,2) blue while the nodes (2,1), (3,1) and (2,3) get the color red. It is not difficult to see that each subgraph induced by the color class is acyclic. Therefore, Theorem 1 implies that the set of observations of Example 1 satisfies CARP.

We remark that Theorem 1, which gives a sufficient condition for CARP, is not a necessary condition. In fact, Example 5, given later in the paper, proves this point. Consequently, there can exist data sets S, with corresponding graphs $G = (V, A)$, which are such that there is no partition of V into two acyclic subgraphs while S satisfies CARP. Or equivalently, there can exist data sets S, with corresponding graphs $G = (V, A)$, that satisfy CARP but such that for any partition of V into two subsets, at least one induced subgraph is cyclic.

Next, we want to note that Cherchye et al. (2007, supplemental material) also generalize the definition of CARP for dealing with households (or groups) of more than two members. Given our construction of the graph, one can in an analogous way extend the above theorem to deal with this general case. That is, if the graph G built from the data S can be node-partitioned into at most M acyclic subgraphs, then there exist $H_0^1, H_0^2, \ldots, H_0^M$ satisfying the corresponding generalization of CARP. Inter alia, this allows us to test for the number of decision makers in the household.

We end this section by showing that the problem of partitioning the nodes of our graph $G = (V, A)$ into two subsets such that each induced subgraph is acyclic, is an NP-complete problem.

Theorem 2 *Given a directed graph $G = (V, A)$ built from the data of a collectively rational consumption behavior problem, deciding whether a node-partitioning of G into two acyclic subgraphs exists, is NP-complete.*

Proof See the Appendix. □

3.3 A Necessary Condition for CARP

In this section, we present a necessary condition for CARP. Consider the data set $S = \{(p_t, q_t); t \in \mathbb{T}\}$. A subset S_1 of S containing at least four observations and at most six observations is called a *double-sum block subset* of S if S_1 contains three pairs of observations (t_1, t_2), (s_1, s_2) and (ℓ_1, ℓ_2) such that $S_1 = \{(p_t, q_t); t \in \{t_1, t_2, s_1, s_2, \ell_1, \ell_2\}\}$ and the following inequalities are met:

N1: $p_{t_i} q_{t_i} > p_{t_i} q_{t_j}$, $p_{s_i} q_{s_i} > p_{s_i} q_{s_j}$ and $p_{\ell_i} q_{\ell_i} > p_{\ell_i} q_{\ell_j}$ with $i \neq j$ and $i, j \in \{1, 2\}$
N2: $p_{t_1} q_{t_1} \geq p_{t_1}(q_{t_2} + q_{s_1})$ if $s_1 \neq t_2$
N3: $p_{t_1} q_{t_1} \geq p_{t_1}(q_{t_2} + q_{\ell_1})$ if $\ell_1 \neq t_2$
N4: $p_{s_1} q_{s_1} \geq p_{s_1}(q_{s_2} + q_{t_1})$ if $t_1 \neq s_2$
N5: $p_{s_1} q_{s_1} \geq p_{s_1}(q_{s_2} + q_{\ell_1})$ if $\ell_1 \neq s_2$
N6: $p_{\ell_1} q_{\ell_1} \geq p_{\ell_1}(q_{\ell_2} + q_{s_1})$ if $s_1 \neq \ell_2$
N7: $p_{\ell_1} q_{\ell_1} \geq p_{\ell_1}(q_{\ell_2} + q_{t_1})$ if $t_1 \neq \ell_2$

The following result shows that if the data set S contains a double-sum block subset, then S does not satisfy CARP.

Theorem 3 *If the data set S contains a double-sum block subset, then S does not satisfy CARP; that is there are no H_0^1 and H_0^2 satisfying Rules 1–3.*

Proof Suppose that the data set S contains a double-sum block subset S_1. This subset has either four, five or six distinct observations.

Case 1: The double-sum block subset S_1 has four distinct observations. Without loss of generality, suppose that $s_1 = t_2 = \ell_2$. Let us assume that there exist hypothetical relations H_0^1 and H_0^2. Suppose without loss of generality that $q_{t_2} H_0^1 q_{s_2}$ and $q_{s_2} H_0^2 q_{t_2}$. The pair of observations (t_1, t_2) is such that either $q_{t_1} H_0^1 q_{t_2}$ or $q_{t_1} H_0^2 q_{t_2}$. If $q_{t_1} H_0^1 q_{t_2}$ then $q_{t_2} H_0^2 q_{t_1}$. The double-sum inequality $p_{t_2} q_{t_2} \geq p_{t_2}(q_{t_1} + q_{s_2})$ (inequality N4) and the fact that $q_{t_1} H_0^1 q_{t_2}$ imply $q_{t_2} H_0^2 q_{s_2}$ from Rule 2b. This leads to $q_{t_2} H_0^1 q_{s_2}$ and $q_{t_2} H_0^2 q_{s_2}$ and Rule 3a is violated. We conclude that $q_{t_1} H_0^2 q_{t_2}$ and thus $q_{t_2} H_0^1 q_{t_1}$. A similar reasoning allows to conclude that $q_{\ell_1} H_0^2 q_{t_2}$ and $q_{t_2} H_0^1 q_{\ell_1}$.
The double-sum inequality $p_{t_1} q_{t_1} \geq p_{t_1}(q_{t_2} + q_{\ell_1})$ from N3 and the fact that $q_{t_2} H_0^1 q_{t_1}$ imply that $q_{t_1} H_0^2 q_{\ell_1}$ (using Rule 2b). The double-sum inequality $p_{\ell_1} q_{\ell_1} \geq p_{\ell_1}(q_{t_1} + q_{t_2})$ from N7 together with $q_{t_1} H_0^2 q_{\ell_1}$ imply, using Rule 2b, that $q_{\ell_1} H_0^1 q_{t_2}$. We then have $q_{\ell_1} H_0^1 q_{t_2}$ and $q_{\ell_1} H_0^2 q_{t_2}$, contradicting Rule

3a. This conclude that if the data set S contains a double-sum block subset with four distinct observations, then there is no H_0^1 and H_0^2 satisfying Rules 1–3.

Case 2: The double-sum block subset S_1 has five distinct observations. Without loss of generality, suppose that $s_1 = t_2$. Suppose also that the hypothetical relations H_0^1 and H_0^2 exist such that $q_{t_2} H_0^1 q_{s_2}$ and $q_{s_2} H_0^2 q_{t_2}$. We know from the reasoning of Case 1 that $q_{t_1} H_0^2 q_{t_2}$ and $q_{t_2} H_0^1 q_{t_1}$.
The double-sum inequality $p_{t_2} q_{t_2} \geq p_{t_2}(q_{s_2} + q_{\ell_1})$ from N_5 together with $q_{s_2} H_0^2 q_{t_2}$ imply, from Rule 2b, that $q_{t_2} H_0^1 q_{\ell_1}$ and $q_{\ell_1} H_0^2 q_{t_2}$ comes from Rule 2a. On the other hand, the double-sum inequality $p_{\ell_1} q_{\ell_1} \geq p_{\ell_1}(q_{\ell_2} + q_{t_2})$ identified by N_6 and $q_{t_2} H_0^1 q_{\ell_1}$ imply that $q_{\ell_1} H_0^2 q_{\ell_2}$ from Rule 2b. Rule 2a allows to conclude $q_{\ell_2} H_0^1 q_{\ell_1}$. The double-sum inequality $p_{t_1} q_{t_1} \geq p_{t_1}(q_{t_2} + q_{\ell_1})$ from N_3 with $q_{t_2} H_0^1 q_{t_1}$ imply that $q_{t_1} H_0^2 q_{\ell_1}$ from Rule 2b. The inequality $p_{\ell_1} q_{\ell_1} \geq p_{\ell_1}(q_{\ell_2} + q_{t_1})$ from N_7 and $q_{\ell_2} H_0^1 q_{\ell_1}$ imply that $q_{\ell_1} H_0^2 q_{t_1}$ from Rule 2b. Therefore, from Rule 2a we have $q_{t_1} H_0^1 q_{\ell_1}$. This implies that $q_{t_1} H_0^1 q_{\ell_1}$ and $q_{t_1} H_0^2 q_{\ell_1}$, hence contradicting Rule 3a. Therefore, there is no H_0^1 and H_0^2 satisfying Rules 1–3.

Case 3: The double-sum block subset S_1 has six distinct observations. A reasoning combining ideas of Case 1 and Case 2 allows to conclude that the existence of a double-sum block subset implies that the data set S does not satisfy CARP.

□

The following example describes a data set S containing a double-sum block subset.

Example 4 Consider a situation with 4 goods ($N = 4$), two household members ($M = 2$) and the following four observed price-quantity combinations ($T = 4$):

1: $q_1 = (8 \quad 2 \quad 2 \quad 0)'$ and $p_1 = (6 \quad 2 \quad 2 \quad 10)'$
2: $q_2 = (1 \quad 8 \quad 3 \quad 0)'$ and $p_2 = (2 \quad 6 \quad 1 \quad 10)'$
3: $q_3 = (1 \quad 2 \quad 8 \quad 0)'$ and $p_3 = (2 \quad 3 \quad 10 \quad 4)'$
4: $q_4 = (1 \quad 2 \quad 0 \quad 5)'$ and $p_4 = (1 \quad 1 \quad 1 \quad 1.7)'$

The inequalities satisfied are:

I_1: $p_s q_s \geq p_s q_t$ for each pair $s, t \in \{1, 2, 3\}$
I_2: $p_3 q_3 > p_3 q_4$ and $p_4 q_4 > p_4 q_3$
I_3: $p_1 q_1 > p_1(q_2 + q_3)$
I_4: $p_2 q_2 > p_2(q_1 + q_3)$
I_5: $p_3 q_3 > p_3(q_1 + q_4)$
I_6: $p_3 q_3 > p_3(q_2 + q_4)$

Consider the subset S_1 of S containing the four observations, grouped in pair of observations as follows. The first pair is (1,3), the second pair is (2,3) and the third pair is (3,4). One can easily check that S_1 satisfies the inequalities N_1–N_7. Therefore, from Theorem 3, the data set S does not satisfy CARP.

Notice that checking whether the data set S contains a double-sum block subset can be done in polynomial time; more precisely, an algorithm for identifying a double-sum block subset can be designed to run in time $O(T^3)$.

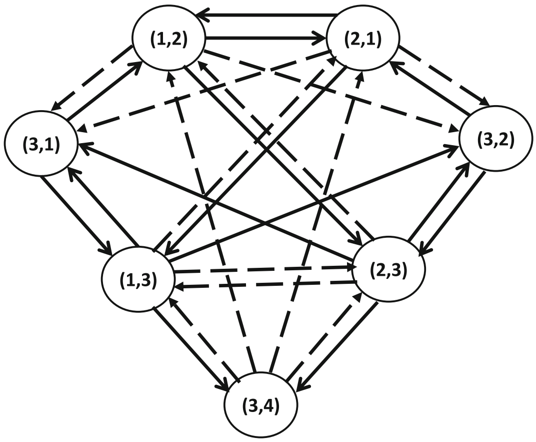


Fig. 2 The graph built from the data of Example 5

Finally, we provide an example of a data set that shows that our necessary and sufficient condition do not coincide. That is, S does not contain a double-sum block subset and the corresponding graph G cannot be node-partitioned into two acyclic subgraphs, yet S satisfies CARP.

Example 5 Consider a situation with 4 goods ($N = 4$), two household members ($M = 2$) and the following four observed price-quantity combinations ($T = 4$):

1: $q_1 = (8 \quad 2 \quad 2 \quad 0)'$ and $p_1 = (6 \quad 2 \quad 2 \quad 10)'$
2: $q_2 = (1 \quad 8 \quad 3 \quad 0)'$ and $p_2 = (2 \quad 6 \quad 1 \quad 10)'$
3: $q_3 = (1 \quad 2 \quad 8 \quad 0)'$ and $p_3 = (2 \quad 3 \quad 10 \quad 4)'$
4: $q_4 = (1 \quad 2 \quad 0 \quad 5)'$ and $p_4 = (1 \quad 1 \quad 1 \quad 1)'$

The inequalities satisfied are:

I_1: $p_s q_s \geq p_s q_t$ for each pair $s, t \in \{1, 2, 3\}$
I_2: $p_3 q_3 > p_3 q_4$
I_3: $p_1 q_1 > p_1(q_2 + q_3)$
I_4: $p_2 q_2 > p_2(q_1 + q_3)$
I_5: $p_3 q_3 > p_3(q_1 + q_4)$
I_6: $p_3 q_3 > p_3(q_2 + q_4)$

The directed graph representation of this problem is given by Fig. 2. We realize that it is not possible to color the nodes of the graph using only two colors in such a way that both subgraphs are acyclic. More explicitly, in any feasible coloring of this graph, one can deduce that nodes (1,3) and (2,3) need to have a different color. It follows that (3,4) cannot be feasibly colored. Further, the data S of this example does not contain a double-sum block subset. However, it is not difficult to see that H_0^1 and H_0^2 defined as follows satisfy Rules 1–3 in Definition 4; i.e., the data S satisfy CARP.

H_0^1: $q_1 H_0^1 q_2, q_1 H_0^1 q_3, q_3 H_0^1 q_2, q_3 H_0^1 q_4$ and $q_s H_0^1 q_s$ for all $s = 1, \ldots, 4$.
H_0^2: $q_2 H_0^2 q_1, q_2 H_0^2 q_3, q_3 H_0^2 q_1, q_3 H_0^2 q_4$ and $q_s H_0^2 q_s$ for all $s = 1, \ldots, 4$.

Notice that the preference relations H_0^1 and H_0^2 obtained for the data of Example 5 have a non-trivial intersection; that is there exist two distinct observations s, t with $p_s q_s \geq p_s q_t$ such that $q_s H_0^1 q_t$ and $q_s H_0^2 q_t$. In the case of Example 5, we have $s = 3$ and $t = 4$. In fact, any pair of hypothetical relations H_0^1 and H_0^2 satisfying CARP for this data will have a non-trivial intersection. This non-trivial intersection is crucial to have data sets as in this example. Indeed, if there exists H_0^1 and H_0^2 with only a trivial intersection for a given data set, then the corresponding graph can be partitioned into two acyclic subgraphs and the converse of Theorem 1 will hold.

In this respect, we can also relate our proposal to the sufficient condition for CARP of Deb (2008b). Essentially, Deb's condition boils down to finding hypothetical relations with a trivial intersection. However, it is fairly easy to verify that Deb's approach, which focuses on a partitioning of some given data set, only considers a subset of all possible hypothetical relations with a trivial intersection. To give a specific illustration, the data in our Example 1 do not satisfy Deb (2008b) condition for the collective model with two household members, while it passes our sufficient condition in Theorem 1. As such, Deb's sufficient condition is more stringent than ours and, in this sense, our condition extends Deb's condition.

4 Heuristics

This section is devoted to the development of simple heuristics for partitioning the directed graph $G = (V, A)$ built in Sect. 3 into two acyclic subgraphs. We first prove that the graph G can always be partitioned into two acyclic subgraphs when $G = G'$ is a line graph. We next present heuristics for solving the general case by combining a greedy rule for coloring the nodes of G with a specific sequence of the nodes. The main motivations to derive simple heuristics are twofold:

1. Sophisticated and time consuming heuristics will not allow the rejection of CARP when they fail to color the nodes of G.
2. Heuristics are used prior to an exact and time consuming algorithm.

The heuristics developed in this section keep as a hard constraint the number of members in the household (two members). This is in accord with practical data where the rationality is tested for two members household data Cherchye et al. (2008); Deb (2008b). Therefore, these heuristics try to color as many nodes as possible with two colors, as opposed to the heuristic developed by Deb (2008b) which aims to color all the nodes using as few colors as possible. Although the heuristics described here focus on partitioning the nodes of the graph G built in Sect. 3, they can easily be adapted to deal with similar graph partitioning problem encountered by Deb (2008b). In what follows, we first prove that a line graph can always be colored using two colors, subsequently we describe heuristics for general case.

Lemma 1 *If $G = G'$ is a line graph, then G can be partitioned into two acyclic subgraphs.*

Proof Since $G = G'$, we have no double sum arcs. As such, the subgraph of G containing nodes (s, t) with $s < t$ is by construction acyclic and the same holds for the subgraph with nodes (s, t) where $s > t$. Clearly these two subgraphs form a partition of G. □

We remark that this special case with $G = G'$ can be quite relevant for empirical exercises; see for instance our own application in Sect. 5. For the general case where G is not a line graph, we develop heuristics by distinguishing coloring strategies on the one hand, and specifying node orderings, or sequences, on the other hand. More specifically, we present four coloring strategies for attempting to color a directed graph into two acyclic subgraphs and 13 sequences of nodes. A heuristic then is a combination of a coloring strategy and an ordering of the nodes.

4.1 Coloring Strategies

CS1: Given a sequence of nodes, color iteratively each node red, unless this would create a red cycle. In case coloring the current node blue would create a blue cycle, we stop (and output: 0), else we color it blue, and continue.

CS2: Given a sequence of nodes, this coloring strategy colors iteratively each even (respectively odd) node red (respectively blue), unless this would create a red (respectively blue) cycle. In case coloring the current node blue (respectively red) would create a blue (respectively red) cycle, we stop (and output: 0), else we color it blue (respectively red), and continue. Notice that in this coloring strategy, a node is called “even” (respectively “odd”) when its position in the sequence is even (respectively odd).

CS3: Given a sequence of nodes, this coloring strategy colors iteratively each node by a randomly generated color (from the set {blue, red}), unless this would create a monochromatic cycle. If coloring the current node red or blue would create a monochromatic cycle, we stop (and output: 0), else we color it with the remaining color, and continue.

CS4: Given a sequence of nodes, this coloring strategy colors iteratively each node with the same color as its predecessor, unless this would create a monochromatic cycle. If coloring the current node with the other color would also create a monochromatic cycle, we stop (and output: 0), else we color it with the other color, and continue.

Notice that in each coloring strategy we often need to check whether a (sub)graph is acyclic. We use the *topological ordering* algorithm to do so, see Ahuja et al. (1993) for more details. This algorithm labels the nodes of the graph ($order(i)$ to each node i) in such a way that every arc joins a lower-labeled node to a higher-labeled node. If for each connected pair of nodes i, j with an arc from i to j we have $order(i) < order(j)$, the graph is acyclic. Otherwise, it contains a cycle. The time complexity of the topological ordering algorithm is $O(m)$ where m is the number of arcs in the graph.

4.2 Ordering of the Nodes

In the previous section, we assumed that a sequence of the nodes was given as input for each of the coloring strategies. Since there are $n!$ possible sequences for a graph G

consisting of n nodes, it is not practical to try all of them. Therefore, we now describe specific sequences of nodes (often based on the structure of the graph) that will be used as input for the above coloring strategies.

Sq1: Sequence 1 is a natural sequence given by: $(0, 1), (0, 2), \ldots, (0, T), (1, 0), (1, 2), \ldots, (1, T), (2, 0), (2, 1), \ldots, (2, T), \ldots, (T - 1, 1), (T - 1, 2), \ldots, (T - 1, T)$ (recall that T is the number of observations). Of course, not all of these nodes need to exist, the non-existing nodes are simply removed from the list.

Sq2: Sequence 2 is the reverse of Sequence 1, hence it starts with $(T - 1, T)$ and ends with (0,1) (provided these nodes exist).

Sq3: Sequence 3 is found by placing each node (s, t) with $s < t$ before each node (s, t) with $s > t$; within each of these two sets of nodes we use the ordering implied by Sequence 1.

Sq4: Sequence 4 is the reverse of Sequence 3. Here, we follow the idea of Sequence 1, but we select node (s, t) with $s > t$ before node (s, t) with $s < t$.

Sq5: In this sequence, the position of a node is chosen randomly.

The next two sequences partition the nodes into those involved in a double-sum inequality, and those that are not. The idea is that nodes involved in a double-sum inequality might be more difficult to color than other nodes, and hence it might be worthwhile to place these nodes in the beginning of the sequence.

Sq6: Sequence 6 also uses the ordering of Sequence 1, but we place each double-sum node before each other node.

Sq7: Sequence 7 is the reverse of Sequence 6.

The following six sequences are based on the degree of a node. The *degree* of a node is the number of arcs it is incident to; the *indegree* is the number of arcs that enter a node while the *outdegree* of a node is the number of arcs that leave a node. Again, the rationale for using this measure is that the number of arcs a node is incident to is a measure of the difficulty of coloring that node.

Sq8: Sequence 8 is found by sorting the nodes with respect to their degree in increasing order; if there is a tie we use the ordering of Sequence 1.

Sq9: Sequence 9 is the reverse of Sequence 8.

Sq10: Sequence 10 is found by sorting the nodes in increasing order of their indegree; if there is a tie we use the ordering of Sequence 1.

Sq11: Sequence 11 is the reverse of Sequence 10.

Sq12: Sequence 12 is found by sorting the nodes in increasing order of their outdegree; if there is a tie we use the ordering of Sequence 1.

Sq13: Sequence 13 is the reverse of Sequence 12.

Notice that we have specified $13 \times 4 = 52$ heuristics since we can combine each of the four coloring strategies with each of the 13 sequences. We will apply all these heuristics on the given instances, and we comment on their efficiency in Sect. 5.2.

We mention that even if the heuristics fail to partition the nodes of G using two colors, its output can be used to reduced the size of the Integer Programming problem to be solved. This could be done by fixing the variables of integer programming model corresponding with the nodes of G colored before the heuristic stops.

5 Computational Experiments

All algorithms have been coded in C using Visual Studio C++ 2005 and are available from the authors upon simple request. The experiments were run on a HP Pavilion dv6000 laptop with AMD Turion (tm) 64 $\times$ 2 Mobile Technology TL-56 processor with 1.80 GHz clock speed and 2047 MB RAM, equipped with Windows Vista. Below, we first provide some details on the data sets used and subsequently, we discuss the computational results.

5.1 Data

Our goal is to investigate the usefulness of the graph construction from Sect. 3, and to assess the quality and the speed of the heuristics proposed in Sect. 4. To do so, we apply the heuristics to two types of data sets drawn from Phase II of the Russian Longitudinal Monitoring Survey, which covers detailed consumption data from a nationally representative sample of Russian two-person households (or couples) during the time period between 1994 and 2003 (Rounds V-XII). When assuming homogeneity of the intra-household allocation process and individual preferences over time, such panel data enable us to treat each household as a time series in its own right. For each household, we focus on a rather detailed consumption bundle that consists of 21 nondurable goods. Only two-person households sharing certain characteristics are retained, which results in a basic sample consisting of 148 couples that are observed eight times. We refer to Cherchye et al. (2008) for more details on the data.

Data I consists of the same real-life instances used by Cherchye et al. (2008); as such this allows us to compare the integer programming approach and the heuristics described here, see Sect. 5.2. In order to obtain bigger data sets that are still usefully interpretable from an economic point of view, these authors merged all households of which males share the same birth year into one data set. In fact, this pertains to testing homogeneity of the intra-household allocation process and individual preferences for these couples. Next, to optimize the CPU times of the Integer Programming approach they applied two efficiency enhancing procedures to minimize the number of observations that need to be considered by their procedures. This resulted in 69 instances with a number of observations that varies between two and 101, for which CARP was tested; for more details, see Cherchye et al. (2008). We refer to this set of instances as Data I.

Second, on the basis of the above sample of 148 households, we also construct 120 synthetic data sets (instances) with varying size; these are contained in Data II. Every synthetic data set is obtained by randomly drawing households from the basic sample. Since each household is observed eight times, data set sizes are multiples of eight and range from eight to 96. As such, we consider data sets with substantially more observations than existing consumer panels; this allows us to analyze in further detail the performance of our heuristics. As far as we know, existing panel data with detailed consumption only contain a rather limited number of observations per household. For example, Christensen (2007) and Blow et al. (2008) use, respectively, Spanish and Danish consumer panels with at most 24 observations per household.

5.2 Computational Results

In this section we discuss the output of the heuristics applied to Data I and Data II.

5.2.1 Data I

The name of the instance is represented by three numbers. The first is the year, the second represents the number of that instance in that year and the last one is the number of observations considered in that instance. Density is the density of the graph given by the percentage of the number of arcs present in that graph divided by the total number of possible arcs.

Table 1 gives the properties of the graph representation of these instances. Notice that each graph contains a cycle. The analysis of this table shows that 57 instances out of 69 lead to a line graph; that is because they have no double-sum arc. This represents more than 82% of the instances, and it clearly shows that it is worthwhile to detect the absence of double-sum arcs in the data: if these arcs are absent one can immediately conclude (usomg Lemma 1 and Theorem 1) that the data satisfy CARP (instead of having to solve an IP-model). The second column of Table 1, entitled "Ref." contains the name which is used to refer to each instance in the rest of this text.

We then apply the heuristics to the remaining 12 instances. Table 2 displays the output of the heuristics. Each column (except for the first two columns and the last column) corresponds to a single instance. The row called "Time" (which corresponds to a specific sequence) reports the CPU time in seconds, which is the average value of the time needed for the four strategies using that particular sequence. The row "CS" identifies the coloring strategies for which we have obtained a partition into acyclic subgraphs. Finally, the last column gives, for each sequence, the total number of strategies for which a feasible coloring was found.

From Table 2, we see that for each instance except I_9(1935-3-101), there is at least one heuristic finding a feasible coloring, meaning that each instance (except I_9) can be partitioned into acyclic subgraphs, and hence, by Theorem 1, satisfies CARP. This shows that (at least for this set of real-life instances) using the graph construction described in Sect. 3 does not lead to a loss of the ability to test whether the data satisfy CARP.

When looking at the results of the heuristics in more detail, we find that strategy 1 and strategy 4 are more successful than the other strategies. In particular, strategy 1 (CS1) is successful (meaning there is a sequence for which a coloring is found) in 11 out of the 12 instances, and strategy 4 (CS4) is successful for ten instances. This contrasts with strategies 2 and 3 which are only successful for two and five instances, respectively. We conclude that when coloring the nodes sequentially, it is better to keep using the same color, and only resort to another color when forced, rather than to build a "balanced" coloring, having approximately the same number of nodes of each color in any partial coloring.

When analyzing the sequences, it can be concluded that the relevance of a particular sequence is limited. Indeed, when a strategy is successful for some instances, there are often (but not always) many sequences for which this strategy is successful. Sequence 6 (Sq6) and Sequence 13 (Sq13) contain the highest number of strategies for which a

Table 1 Properties of the Graph representation of the instances of Data I

Instance	Ref.	Obser.	Nodes	double-sum	Simple arcs	DS arcs	DS nodes	Total arcs	Density	Cyclic
1918-1-3	–	3	5	0	8	0	0	8	40.00	1
1924-1-2	–	2	2	0	2	0	0	2	100.00	1
1924-2-2	–	2	2	0	2	0	0	2	100.00	1
1924-3-7	I_1	7	22	2	52	1	2	53	11.47	1
1924-4-15	I_2	15	95	5	511	2	3	513	5.74	1
1926-1-2	–	2	2	0	2	0	0	2	100.00	1
1926-2-2	–	2	2	0	2	0	0	2	100.00	1
1926-3-3	–	3	5	0	8	0	0	8	40.00	1
1926-4-11	I_3	11	48	4	167	2	4	169	7.49	1
1927-1-3	–	3	5	0	8	0	0	8	40.00	1
1927-2-4	–	4	8	0	14	0	0	14	25.00	1
1927-3-4	–	4	7	0	13	0	0	13	30.95	1
1927-4-12	I_4	12	68	42	280	17	17	297	6.52	1
1927-5-27	I_5	27	279	590	1951	61	34	2012	2.59	1
1928-1-2	–	2	2	0	2	0	0	2	100.00	1
1928-2-7	–	7	23	0	60	0	0	60	11.86	1
1929-1-3	–	3	5	0	8	0	0	8	40.00	1
1929-2-3	–	3	5	0	8	0	0	8	40.00	1
1929-3-5	–	5	12	0	29	0	0	29	21.97	1
1929-4-32	I_6	32	410	447	3639	21	27	3660	2.18	1
1930-1-2	–	2	2	0	2	0	0	2	100.00	1
1930-2-2	–	2	2	0	2	0	0	2	100.00	1
1930-3-6	–	6	21	0	63	0	0	63	15.00	1
1930-4-16	I_7	16	118	30	682	17	15	699	5.06	1
1930-5-17	I_8	17	139	11	976	9	14	985	5.14	1
1931-1-2	–	2	2	0	2	0	0	2	100.00	1
1931-2-2	–	2	2	0	2	0	0	2	100.00	1
1932-1-2	–	2	2	0	2	0	0	2	100.00	1
1932-2-5	–	5	12	0	23	0	0	23	17.42	1
1932-3-6	–	6	19	0	60	0	0	60	17.54	1
1933-1-4	–	4	9	0	19	0	0	19	26.39	1
1935-1-2	–	2	2	0	2	0	0	2	100.00	1
1935-2-7	–	7	22	0	61	0	0	61	13.20	1
1935-3-101	I_9	101	4384	46916	121269	3052	2672	124321	0.65	1
1936-1-2	–	2	2	0	2	0	0	2	100.00	1
1936-2-2	–	2	2	0	2	0	0	2	100.00	1
1936-3-2	–	2	2	0	2	0	0	2	100.00	1
1936-4-2	–	2	2	0	2	0	0	2	100.00	1
1936-5-5	–	5	11	0	25	0	0	25	22.73	1

Table 1 continued

Instance	Ref.	Obser.	Nodes	double-sum	Simple arcs	DS arcs	DS nodes	Total arcs	Density	Cyclic
1936-6-40	I_{10}	40	755	1121	10049	64	46	10113	1.78	1
1937-1-2	–	2	2	0	2	0	0	2	100.00	1
1937-2-4	–	4	9	0	19	0	0	19	26.39	1
1937-3-5	–	5	13	0	30	0	0	30	19.23	1
1937-4-21	I_{11}	21	226	111	1953	26	19	1979	3.89	1
1938-1-2	–	2	2	0	2	0	0	2	100.00	1
1938-2-4	–	4	8	0	15	0	0	15	26.79	1
1938-3-4	–	4	8	0	14	0	0	14	25.00	1
1938-4-6	–	6	17	0	43	0	0	43	15.81	1
1938-5-9	–	9	39	0	129	0	0	129	8.70	1
1938-6-16	–	16	108	0	511	0	0	511	4.42	1
1939-1-2	–	2	2	0	2	0	0	2	100.00	1
1940-1-2	–	2	2	0	2	0	0	2	100.00	1
1940-2-2	–	2	2	0	2	0	0	2	100.00	1
1940-3-3	–	3	5	0	8	0	0	8	40.00	1
1940-4-18	–	18	141	0	852	0	0	852	4.32	1
1941-1-2	–	2	2	0	2	0	0	2	100.00	1
1941-2-3	–	3	4	0	5	0	0	5	41.67	1
1941-3-26	I_{12}	26	294	257	2353	74	66	2427	2.82	1
1945-1-2	–	2	2	0	2	0	0	2	100.00	1
1945-2-2	–	2	2	0	2	0	0	2	100.00	1
1948-1-2	–	2	2	0	2	0	0	2	100.00	1
1948-2-4	–	4	7	0	10	0	0	10	23.81	1
1948-3-4	–	4	8	0	15	0	0	15	26.79	1
1949-1-2	–	2	2	0	2	0	0	2	100.00	1
1950-1-5	–	5	12	0	25	0	0	25	18.94	1
1954-1-2	–	2	2	0	2	0	0	2	100.00	1
1954-2-2	–	2	2	0	2	0	0	2	100.00	1
1962-1-2	–	2	2	0	2	0	0	2	100.00	1
1962-2-3	–	3	5	0	8	0	0	8	40.00	1

feasible coloring was found, making them the most attractive sequences. In particular, the heuristic obtained by combining sequence 6 (Sq6) and strategy 1 (CS1) is very successful indeed: it solves all the instances except the one that is not solved by any heuristic (I_9).

In fact, instance I_9 is a particular instance in the sense that it is the only instance that was not solved by the IP-model of Cherchye et al. (2008) after one hour of computing time. Our best heuristic (combining strategy 1 and sequence 6) led to a partial feasible coloring of 4224 nodes, i.e., about 95% of the nodes. We also verified that this instance passes our necessary condition in Theorem 3. On the other hand, we

Table 2 Output of heuristics for instances of Data I

Instance		I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	Total
Sq1	Time	0.00	0.00	0.00	0.00	0.02	0.12	0.00	0.00	34.04	0.73	0.03	0.08	16
	CS	1,2,3,4	1,4	1,4	4	–	–	1,4	4	–	4	1	1,4	
Sq2	Time	0.00	0.00	0.00	0.00	0.09	0.23	0.00	0.00	78.54	1.10	0.04	0.01	14
	CS	1,3,4	1,4	1,2,4	1,4	–	1	1	4	–	–	1	1	
Sq3	Time	0.00	0.00	0.00	0.00	0.01	0.16	0.00	0.00	31.33	0.70	0.04	0.01	16
	CS	1,2,3,4	1,4	1,3,4	4	–	–	1	4	–	–	1,4	1,4	
Sq4	Time	0.00	0.00	0.00	0.00	0.09	0.30	0.00	0.01	123.93	1.35	0.05	0.01	12
	CS	1,4	1,4	1,4	1	–	1	1	–	–	–	1,4	1	
Sq5	Time	0.00	0.00	0.00	0.00	0.03	0.20	0.00	0.00	12.25	1.38	0.01	0.04	5
	CS	1,4	–	1,2	–	–	–	–	1	–	–	–	–	
Sq6	Time	0.00	0.00	0.00	0.00	0.09	0.21	0.00	0.00	6.00	1.54	0.03	0.07	22
	CS	1,2,3,4	1,4	1,3,4	1,2,3	1,4	1	1	1,4	–	1,4	1	1	
Sq7	Time	0.00	0.00	0.00	0.00	0.02	0.12	0.00	0.00	34.12	0.73	0.02	0.08	16
	CS	1,2,3,4	1,4	1,4	4	–	–	1,4	4	–	4	1	1,4	
Sq8	Time	0.00	0.00	0.00	0.00	0.14	0.50	0.01	0.01	788.55	3.15	0.06	0.20	6
	CS	1,4	1	1,4	–	–	1	–	–	–	–	–	–	
Sq9	Time	0.00	0.00	0.00	0.00	0.05	0.14	0.00	0.00	22.54	1.04	0.00	0.04	15
	CS	1,2,3,4	1	1,2	1,4	1	1	1	1	–	1	–	3	
Sq10	Time	0.00	0.00	0.00	0.00	0.06	0.22	0.00	0.00	27.77	1.50	0.05	0.09	7
	CS	1,4	1	1	1,4	–	1	–	–	–	–	–	–	
Sq11	Time	0.00	0.00	0.00	0.00	0.04	0.25	0.00	0.01	6.09	0.88	0.02	0.05	18
	CS	1,2,3,4	1,4	1,2,3,4	–	–	1,4	1	1,4	–	1	1	1	
Sq12	Time	0.00	0.00	0.00	0.00	0.09	0.30	0.01	0.01	363.63	1.71	0.04	0.10	6
	CS	1	1	1	–	–	1	–	1	–	1	–	–	
Sq13	Time	0.00	0.00	0.00	0.00	0.08	0.24	0.00	0.00	17.65	0.95	0.01	0.05	21
	CS	1,2,3,4	1,4	1,2,3,4	1,4	1,4	1	1,3,4	1	–	1	–	1	

find that this graph can be colored by the heuristics using three colors. Very recently, Talla Nobibon et al. (2009) showed that this particular graph can, in fact, be colored using two colors, using a dedicated enumerative algorithm. This result enforces the usefulness of Theorem 1.

Table 2 also shows that the heuristics are quite fast. Computation time for most instances are within 0.1 second, and always (except for I_9) within two seconds. This is in contrast with the computation time of Cherchye et al. (2008), who report computation times up to five minutes for their instances. It should be noted, though, that solving the IP-model can lead to a conclusive answer, while the possible failure of a heuristic to produce a coloring gives no information about whether the data satisfy CARP. Nonetheless, we conclude that investing a little computation time to test for CARP quickly is a sensible approach for real life data (Data I).

Table 3 Properties of the Graph representation of the instances of Data II

Instance	Ref.	Obser.	Nodes	double-sum	Simple arcs	DS arcs	DS nodes	Total arcs	Density	Cyclic
Rand-1	R_1	8	25	20	45	1	2	46	6.96	1
Rand-2	R_2	16	108	285	445	12	13	457	3.87	5
Rand-3	R_3	24	254	1098	1651	46	46	1697	2.63	8
Rand-4	R_4	32	449	2652	3920	113	105	4033	2.00	10
Rand-5	R_5	40	718	5608	8079	141	133	8220	1.59	10
Rand-6	R_6	48	1023	9420	13671	212	188	13883	1.32	9
Rand-7	R_7	56	1406	16003	22452	494	369	22946	1.15	10
Rand-8	R_8	64	1808	23199	31981	456	351	32437	0.97	10
Rand-9	R_9	72	2276	32338	44917	710	567	45627	0.88	10
Rand-10	R_{10}	80	2845	45464	63526	835	654	64361	0.79	10
Rand-11	R_{11}	88	3448	59654	84963	1112	838	86075	0.72	10
Rand-12	R_{12}	96	4045	73973	106191	1065	832	107256	0.66	10

5.2.2 Data II

The name of a group of instances is represented by "Rand" followed by a number. Each group contains ten randomly generated instances. Rand is used to express the random characteristics of these instances and the number refers to the number of instances with eight observations aggregated. For instance, Rand-5 has $8 \times 5 = 40$ observations as it is the aggregation of five instances, each with eight observations.

Table 3 gives the properties of the graph representation of the instances in Data II. In this table, each entry (except the entries in the last column) represents the average value of the ten values obtained for each instance in that group. In the last column (Cyclic), we give the number of instances in that group that contain both a cycle and a double-sum arc. Therefore, instances with only a cycle and no double-sum arc are not counted, since these are solved by Lemma 1.

Table 4 displays the output of the heuristics when applied to the instances in Data II. The notations are the same as in Table 2; an entry in the row "CS" is a 4-tuple indicating the number of instances solved by CS1, CS2, CS3, and CS4 respectively. Notice however that here an entry in the row "Time" is the average over the ten values obtained for the instances in that group. The last row of Table 4 (Nr. solved) reported the number of instances in each group for which the heuristics are able to find an optimal partition.

When analyzing the results of Table 4, we see that for the instances with at most 40 observations, the heuristics behave excellent. In fact, for each instance, the heuristics found an acyclic partition. Moreover, the CPU time used by the heuristics is less than two seconds. These observations confirm the results from Data I.

When the number of observations grows, the effectiveness of the heuristics drops. This is clearly seen from the last row of Table 4. Still, more than 60% of the instances whose number of observations is between 48 and 72 are solved in a reasonable amount of time (less than a minute). However, when the number of observations further increases, the effectiveness of the heuristic goes further down. We recall that there are

Table 4 Output of heuristics for instances of Data II

Instance		R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}	R_{12}
Sq1	Time	0.00	0.01	0.08	0.35	1.38	4.53	6.60	18.17	19.97	34.51	38.98	56.26
	CS	1,0,0,1	5,1,1,5	7,3,2,7	7,1,1,9	6,0,0,8	5,1,0,7	4,0,0,6	4,0,1,4	1,0,0,3	0,0,0,2	0,0,0,1	0,0,0,1
Sq2	Time	0.00	0.01	0.08	0.35	1.51	3.77	9.12	18.89	25.58	38.11	52.29	80.11
	CS	1,0,0,1	5,3,1,4	7,2,2,7	7,1,1,7	6,2,1,6	4,0,1,4	4,0,0,1	4,0,0,3	0,0,0,2	0,0,0,0	0,0,0,0	0,0,0,0
Sq3	Time	0.00	0.01	0.08	0.37	1.51	4.69	7.58	19.24	24.21	42.03	58.57	92.67
	CS	1,0,0,1	5,1,1,5	7,3,1,6	7,2,2,6	6,0,0,4	5,1,0,4	4,0,0,3	4,0,0,3	1,0,0,1	0,0,0,1	0,0,0,0	0,0,0,0
Sq4	Time	0.00	0.01	0.09	0.38	1.66	4.21	10.19	21.13	31.06	46.70	88.99	136.29
	CS	1,1,1,1	5,1,2,5	7,2,3,7	7,3,1,5	6,0,2,5	4,0,0,3	4,0,0,0	4,0,0,1	0,0,0,2	0,0,0,0	0,0,0,0	0,0,0,0
Sq5	Time	0.00	0.01	0.10	0.37	1.26	3.26	3.35	11.20	12.98	14.81	23.93	21.77
	CS	1,0,1,0	3,1,2,1	7,3,4,3	6,1,2,4	2,1,0,0	2,0,0,0	1,0,0,0	1,0,0,0	0,0,0,0	0,0,0,0	0,0,0,0	0,0,0,0
Sq6	Time	0.00	0.01	0.08	0.33	1.32	2.99	3.87	16.99	33.88	10.54	21.67	0.80
	CS	1,1,1,1	4,4,4,4	7,5,6,6	8,5,5,5	8,3,3,3	5,2,2,2	3,1,1,1	5,2,1,4	5,2,1,3	0,0,1,2	1,0,0,0	0,0,0,0
Sq7	Time	0.00	0.01	0.08	0.35	1.37	4.52	6.59	18.22	19.97	34.43	38.95	56.47
	CS	1,0,0,1	5,1,1,5	7,3,2,7	7,1,1,9	6,0,0,8	5,1,0,7	4,0,0,6	4,0,1,4	1,0,0,3	0,0,0,2	0,0,0,1	0,0,0,1
Sq8	Time	0.00	0.01	0.14	0.73	3.41	10.23	25.40	56.35	100.94	164.76	347.73	537.32
	CS	0,0,0,0	3,1,1,3	5,2,3,3	2,2,1,3	2,0,0,1	2,0,1,1	1,0,0,0	1,0,0,0	0,0,0,0	0,0,0,0	0,0,0,0	0,0,0,0
Sq9	Time	0.00	0.01	0.12	0.37	1.55	4.07	2.32	15.58	28.18	13.63	6.05	4.55
	CS	1,1,1,1	2,3,2,2	7,7,6,6	6,4,3,6	6,3,5,4	4,2,3,3	1,0,0,1	4,1,0,1	2,1,0,3	1,0,0,0	0,0,0,0	0,0,0,0
Sq10	Time	0.00	0.01	0.09	0.39	1.46	4.77	9.18	18.99	29.85	61.56	105.78	132.23
	CS	0,0,0,0	2,2,2,2	5,3,2,5	4,1,1,4	2,0,1,1	2,0,0,3	1,0,0,0	2,0,0,0	0,0,0,0	0,0,0,0	0,0,0,0	0,0,0,0
Sq11	Time	0.00	0.01	0.07	0.30	1.21	2.64	1.80	10.29	13.41	5.41	1.33	4.53
	CS	1,1,1,1	2,2,3,2	6,5,4,6	6,3,4,6	6,3,3,5	4,2,2,2	1,1,0,1	4,1,1,1	1,0,1,1	0,0,0,1	0,0,0,0	0,0,0,0

Table 4 continued

Instance		R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}	R_{12}
Sq12	Time	0.00	0.00	0.10	0.42	1.39	4.35	7.36	16.45	29.42	44.76	51.29	96.40
	CS	0,1,1,0	3,2,1,1	4,2,2,4	2,1,2,3	2,1,0,1	2,0,0,0	1,0,0,0	1,0,0,0	0,0,0,0	0,0,0,0	0,0,0,0	0,0,0,0
Sq13	Time	0.00	0.02	0.11	0.46	2.18	4.88	7.47	22.74	40.05	39.15	82.32	92.34
	CS	1,1,1,1	4,4,4,2	7,6,6,7	7,6,4,7	8,5,5,6	5,3,2,2	2,0,1,4	4,1,0,3	3,3,1,2	1,0,0,1	0,0,0,0	0,0,0,0
Nr. solved		10	10	10	10	10	8	6	7	6	3	2	1

three possible explanations for this: (i) either a coloring exists, but the heuristics fail to find one, or (ii) the graph does not admit a coloring in spite of the fact that the data satisfy CARP, or (iii) the data simply does not satisfy CARP. More sophisticated heuristics might shed a light on this question.

Overall, Table 4 reports that 83 instances out of 120 are solved using the heuristics; that is around 69% of the instances. The findings obtained after the application of heuristics to the instances in Data I are confirmed here. For instance, sequence 6 (Sq6) and sequence 13 (Sq13) are still the most attractive sequences, while coloring strategies 1 (CS1) and 4 (CS4) are the most successful strategies.

Summarizing, the computational results suggest that

1. verifying whether the graph derived from the data contains double-sum arcs is rewarding for real life instances,
2. the graph construction from Sect. 3 is useful for testing CARP at least for medium-sized instances (up to 75 observations), and
3. investing a little computation time (two seconds) trying to find a heuristic coloring often prevents the usage of a much more time-demanding exact algorithm.

6 Summary and Conclusions

In this paper, we consider the computational problem of testing whether observed data from household consumption behavior satisfies the Collective Axiom of Revealed Preferences (CARP). We construct a directed graph from the data of observed household consumption which is such that the existence of a node-partitioning giving rise to two induced subgraphs that are acyclic implies that the data satisfies CARP. We also propose a necessary condition for CARP. We provide an example showing that there exists data which do not satisfy the necessary condition and the corresponding graph does not admit a partition into two acyclic subgraphs while the data satisfies CARP. Although checking the necessary condition can be achieved in polynomial time, we prove that partitioning the nodes of the obtained graph into two acyclic subgraphs is an NP-complete problem.

We prove that when the graph is a line graph, the data used to build that graph satisfies CARP. For graphs that do contain double-sum arcs, we propose and implement heuristics which are quite fast and can be used to check large data sets for CARP. The heuristics proposed are used prior to an exact and time consuming algorithm. Moreover, if the outcome of the heuristics is not conclusive, it can be used to reduced the size of the IP model to be solved. Applied to real life and synthetic data, the heuristics turn out to be very effective for testing CARP. Moreover, the running time of the heuristics are usually within a fraction of second.

An important research direction that might be pursued in the future is an attempt to fill the gap between the necessary and the sufficient conditions proposed in this paper by providing stronger conditions. Since most data satisfying CARP leads to graph that can be partitioned into two acyclic subgraphs, a second obvious extension that deserves attention is the development of an exact algorithm for partitioning the nodes of a given directed graph into two acyclic subgraphs.

Appendix: Proof of Theorem 2

Proof The proof is a refinement of Deb's proof (2008b) for arbitrary graphs G to our special case. It uses the Not-All-Equal-3Sat problem defined as follows.
INSTANCE: Set $X = \{x_1, \ldots, x_n\}$ of n variables, collection $C = \{C_1, \ldots, C_m\}$ of m clauses over X such that each clause $C_l = x_i \vee x_j \vee x_k$ depends on exactly three distinct variables.
QUESTION: Is there a truth assignment for C such that each clause in C has at least one true literal and at least one false literal?
Garey and Johnson (1979) proved that the Not-All-Equal-3Sat problem is NP-complete.

For a given instance of the Not-All-Equal-3Sat problem, consider the following polynomial time reduction to an instance of our graph partitioning problem. For each variable $x_i \in X$, we have a pair of observations, that gives rise to the existence of two nodes called $(x_i, \bar{x}_i)$ and $(\bar{x}_i, x_i)$. (Notice that the existence of these nodes has implications for the prices and the quantities of goods corresponding to those observations. Here, we will ignore this issue, and simply create nodes assuming that the prices and quantities satisfy the corresponding relationships.) Hence, if $|X| = n$, we have $2n$ such nodes called *variable nodes* as they come from variables. For each clause $C_l = x_i \vee x_j \vee x_k \in C$, we define 18 *clause nodes* as follows. There are three *initial nodes* (x_i^l, x_j^l), (x_j^l, x_k^l) and (x_k^l, x_i^l) and there are three *complement nodes* (x_j^l, x_i^l), (x_k^l, x_j^l) and (x_i^l, x_k^l). Moreover, for each initial node, we define four *path nodes* which are used to create a path from that initial node to a given variable node. We say that these four path nodes are *associated* to this initial node. Explicitly, for the first initial node (x_i^l, x_j^l), we have $(s^l, \bar{x}_i)$, $(\bar{x}_i, s^l)$, (s^l, x_j^l) and (x_j^l, s^l); we refer to these four path nodes as the first, the second, the third and the fourth path nodes. For the second initial node (x_j^l, x_k^l), we define $(t^l, \bar{x}_j)$, $(\bar{x}_j, t^l)$, (t^l, x_k^l) and (x_k^l, t^l). Finally, for the third initial node (x_k^l, x_i^l), are created the path nodes $(u^l, \bar{x}_k)$, $(\bar{x}_k, u^l)$, (u^l, x_i^l) and (x_i^l, u^l). For each initial node, we define the path containing the nodes from the first path node to the complement node via the initial node. For instance, for the initial node (x_i^l, x_j^l), we have the path $P(x_i^l, x_j^l) = \{(s^l, \bar{x}_i), (\bar{x}_i, s^l), (s^l, x_j^l), (x_j^l, s^l), (x_i^l, x_j^l), (x_j^l, x_i^l)\}$. We use P to denote such path. In total, we have $|V| = 2n + 18m$ nodes. To complete the definition of our graph $G = (V, A)$, we now specify the arcs. Clearly, as described in Sect. 3, there is an arc directed from (u, v) to (v, t) whenever (u, v) and (v, t) are nodes in V. Also, we add specific double-sum arcs. These arcs are derived from specific double sum inequalities. For a given clause $C_l = x_i \vee x_j \vee x_k \in C$, we consider 9 double sum inequalities, 3 for each initial node. For the initial node (x_i^l, x_j^l), we have three inequalities:

1. $p_{x_j^l} q_{x_j^l} \geq p_{x_j^l}(q_{x_i^l} + q_{s^l})$. This inequality implies the existence of arcs from node (x_j^l, s^l) to nodes $(x_i^l, .)$, and arcs from node (x_j^l, x_i^l) to nodes $(s^l, .)$. Notice that all these double sum arcs are between clause nodes from the clause C_l.
2. $p_{s^l} q_{s^l} \geq p_{s^l}(q_{x_j^l} + q_{\bar{x}_i})$. This inequality implies the existence of double sum arcs from node $(s^l, \bar{x}_i)$ to nodes $(x_j^l, .)$, and from node (s^l, x_j^l) to nodes $(\bar{x}_i, .)$. Notice that there may be an arc between two nodes of different clauses; indeed, if x_i

occurs in another clause C_r, then there is a double sum arc from (s^l, x_j^l) to node $(\bar{x}_i, s^r)$.

3. $p_{\bar{x}_i} q_{\bar{x}_i} \geq p_{\bar{x}_i}(q_{x_i} + q_{s_l})$. This inequality implies the existence of arcs from node $(\bar{x}_i, s^l)$ to nodes $(x_i, .)$, and from node $(\bar{x}_i, x_i)$ to nodes $(s^l, .)$. Again, if $\bar{x}_i$ occurs in another clause C_r, then there is an arc from $(\bar{x}_i, s^l)$ to node (x_i, s^r).

For each of the two remaining initial nodes (x_j^l, x_k^l) and (x_k^l, x_i^l), the construction is similar. We simply list here the corresponding double sum inequalities. For the initial node (x_j^l, x_k^l), we have the three inequalities

$$4. p_{x_k^l} q_{x_k^l} \geq p_{x_k^l}(q_{x_j^l} + q_{t^l}) \quad 5. p_{t^l} q_{t^l} \geq p_{t^l}(q_{x_k^l} + q_{\bar{x}_j}) \quad 6. p_{\bar{x}_j} q_{\bar{x}_j} \geq p_{\bar{x}_j}(q_{x_j} + q_{t_l}),$$

and for the initial node (x_k^l, x_i^l), the double sum inequalities are:

$$7. p_{x_i^l} q_{x_i^l} \geq p_{x_i^l}(q_{x_k^l} + q_{u^l}) \quad 8. p_{u^l} q_{u^l} \geq p_{u^l}(q_{x_i^l} + q_{\bar{x}_k}) \quad 9. p_{\bar{x}_k} q_{\bar{x}_k} \geq p_{\bar{x}_k}(q_{x_k} + q_{u_l}).$$

This completes the definition of our graph. Clearly, the above reduction can be done in polynomial time. Notice that each consecutive pair of nodes in each path P induces a cycle.

To have an overview of the above reduction, let us consider the following example. $X = \{x, y, z\}$ and there are two clauses $C_1 = x \vee y \vee z$ and $C_2 = \neg x \vee y \vee \neg z$. Remark that the assignment $x = y = 1$ and $z = 0$ is a solution to this Not-All-Equal-3Sat problem. From our reduction, $V = \{(x, \neg x), (\neg x, x), (y, \neg y), (\neg y, y), (z, \neg z), (\neg z, z), (x^1, y^1), (y^1, x^1), (y^1, s^1), (s^1, y^1), (\neg x, s^1), (s^1, \neg x), (y^1, z^1), (z^1, y^1), (z^1, t^1), (t^1, z^1), (\neg y, t^1), (t^1, \neg y), (z^1, x^1), (x^1, z^1), (x^1, u^1), (u^1, x^1), (\neg z, u^1), (u^1, \neg z), (\neg x^2, y^2), (y^2, \neg x^2), (y^2, s^2), (s^2, y^2), (x, s^2), (s^2, x), (y^2, \neg z^2), (\neg z^2, y^2), (\neg z^2, t^2), (t^2, \neg z^2), (\neg y, t^2), (t^2, \neg y), (\neg z^2, \neg x^2), (\neg x^2, \neg z^2), (\neg x^2, u^2), (u^2, \neg x^2), (z, u^2), (u^2, z)\}$. The graph obtained is depicted in Fig. 3. Notice that for reason of clarity, not all the double sum arcs are present in that figure.

Now, we prove that the graph $G = (V, A)$ obtained by the reduction can be partitioned into two acyclic subgraphs if and only if the instance of the Not-All-Equal-3Sat problem is a Yes-instance.

On one hand, if graph G can be partitioned into two acyclic subgraphs G_1 and G_2, then for each variable $x_i \in X$, if the node $(x_i, \bar{x}_i) \in G_1$, then we set the variable $x_i = 1$; else we set the variable $x_i = 0$. Let us prove that this assignment is a truth assignment for the set of clauses C. Let $C_l = x_i \vee x_j \vee x_k \in C$ be any clause $1 \leq l \leq m$. If $x_i = x_j = x_k = 1$ or $x_i = x_j = x_k = 0$, then the nodes (x_i^l, x_j^l), (x_j^l, x_k^l) and (x_k^l, x_i^l) are in the same partition and this will contradict the fact that each subgraph is acyclic.

On the other hand, if there is a truth assignment for C, then consider the following partition of G. For each $x_i \in X$, if $x_i = 1$ we color the variable node $(x_i, \bar{x}_i)$ red and $(\bar{x}_i, x_i)$ blue. Otherwise, if $x_i = 0$ we color the variable node $(x_i, \bar{x}_i)$ blue and $(\bar{x}_i, x_i)$ red. Moreover, we alternate the color of the nodes on the path P by coloring the first path node different from the corresponding variable node. This completes the coloring.

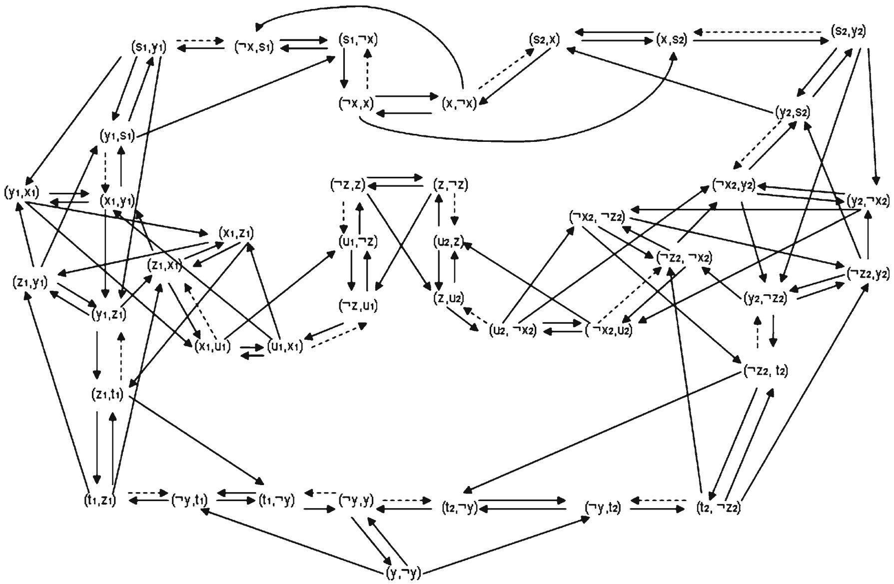


Fig. 3 Example of reduction

Clearly, the blue subgraph and the red subgraph define a partition of G. It remains to show that each subgraph is acyclic. We associate a *parity* to each node (except variable nodes) as follows: the first path node, the third path node, and the corresponding initial node are *odd* nodes, while the second path node, the fourth path node, and the complement node of the corresponding initial node are *even* nodes. Let us now argue that each cycle in G is not monochromatic. First, we consider cycles containing nodes from different clauses.

As described above, some double sum arcs may link path nodes of different clauses. From the definition of our coloring, it turns out that such an arc links nodes of different colors. In fact, suppose that there is a double sum arc from (s^l, x^l_j) to a path node $(\bar{x}_j, s^r)$ of another clause C_r. Then the coloring implies that $(\bar{x}_j, s^r)$ and $(\bar{x}_j, s^l)$ have the same color. Therefore, $(\bar{x}_j, s^r)$ and (s^l, x^l_j) have different colors. Thus, any cycle including nodes from different clauses linked using a double sum arc, is not monochromatic. It follows that any monochromatic cycle containing nodes of different clauses necessarily contains a variable node. Moreover, since each arc leaving a variable node goes to a node with a different color, any cycle containing a variable node is not monochromatic. We conclude that cycles with clause nodes from different clauses are not monochromatic.

Second, we consider cycles within the subgraph defined by a single clause. Obviously, no monochromatic cycle can contain an arc between two consecutive nodes from path P. Thus each cycle in the subgraph consists of three arcs, linking three nodes of the three different paths that exist within each subgraph.

We claim that there do not exist arcs between nodes of different parity.

This claim implies that a monochromatic cycle would consist of three nodes of the same parity. However, the three initial nodes have the same parity, and the solution of the Not-All-Equal-3Sat problem implies that these nodes do not form a monochromatic cycle. The coloring then implies that any set of three nodes of the same parity do not form a monochromatic cycle. Hence, the validity of our claim implies the result.

To establish the claim, observe that each regular (i.e., non double sum) arc between nodes of different paths is induced by a literal from the initial nodes, e.g. from $(., x_i^l)$ to $(x_i^l, .)$. Since this literal occurs in the three nodes once in the first position and once in second position, this implies that each regular arc links nodes of the same parity. In fact, it can be verified that this is also true for double sum arcs. Hence, the claim is valid. This completes the proof. □

Acknowledgements We thank the editor Hans Amman and an anonymous referee for helpful comments. Laurens Cherchye gratefully acknowledges financial support from the Research Fund K.U. Leuven through the grant STRT1/08/004.

References

Afriat, S. (1967). The construction of utility functions from expenditure data. *International Economic Review, 8*, 67–77.

Ahuja, R., Magnanti, T., & Orlin, J. (1993). *Network flows: Theory, algorithms, and applications*. Upper Saddle River, NJ: Prentice Hall.

Blow, L., Browning, M., & Crawford, I. (2008). Revealed preference analysis of characteristic models. *Review of Economic Studies, 75*, 371–389.

Brown, D., & Matzkin, R. (1996). Testable restrictions on the equilibrium manifold. *Econometrica, 64*, 1249–1262.

Brown, D., & Shannon, C. (2000). Uniqueness, stability and comparative statics in rationalizable walrasian markets. *Econometrica, 68*, 1529–1540.

Browning, M., & Chiappori, P. (1998). Efficient intra-household allocations: A general characterization and empirical tests. *Econometrica, 68*, 1241–1278.

Carvajal, A., Ray, I., & Snyder, S. (2004). Equilibrium behavior in markets and games: Testable restrictions and identification. *Journal of Mathematical Economics, 40*, 1–40.

Chang, G., Chen, C., & Chen, Y. (2004). Vertex and tree arboricities of graphs. *Journal of Combinatorial Optimization, 8*, 295–306.

Chen, Z. (2000). Efficient algorithm for acyclic colorings of graphs. *Theoretical Computer Science, 230*, 75–95.

Cherchye, L., Crawford, I., De Rock, B., & Vermeulen, F. (2009). The revealed preference approach to demand. Chapter 9 in quantifying consumer preferences: Estimating demand systems. In D. Slottje, *Contributions to economic analysis*. UK: Emerald Press.

Cherchye, L., De Rock, B., Sabbe, J., & Vermeulen, F. (2008). Nonparametric tests of collectively rational consumption behavior: An integer programming procedure. *Journal of Econometrics, 147*, 258–265.

Cherchye, L., De Rock, B., & Vermeulen, F. (2007). The collective model of household consumption: A nonparametric characterization. *Econometrica, 75*, 553–574.

Cherchye, L., De Rock, B., & Vermeulen, F. (2008). Analyzing cost efficient production behavior under economies of scope: A nonparametric methodology. *Operations Research, 56*, 204–221.

Cherchye, L., De Rock, B., & Vermeulen, F. (2010). An Afriat theorem for the collective model of household consumption. *Journal of Economic Theory, 145*, 1142–1163.

Chiappori, P. (1988). Rational household labor supply. *Econometrica, 56*, 63–89.

Chiappori, P. (1992). Collective labor supply and welfare. *Journal of Political Economy, 100*, 437–467.

Chiappori, P., & Ekeland, I. (2006). The micro economics of group behavior: General characterization. *Journal of Economic Theory, 130*, 1–26.

Chiappori, P., & Ekeland, I. (2009). The micro economics of efficient group behavior: Identification. *Econometrica, 77*, 763–800.

Christensen, M. (2007). Integrability of demand accounting for unobservable heterogeneity: A test on panel data. *IFS Working paper W14/07*, University of Manchester.

Chung-Piaw, T., & Vohra, R. (2003). Afriat's theorem and negative cycles, mimeo, Northwestern University.

Deb, R. (2008). *Acyclic partitioning problem is NP-complete for* $k = 2$. *Private communication*. United States: Yale University.

Deb, R. (2008b). An efficient nonparametric test of the collective household model, *Working paper*. Yale University, United States.

Diewert, E. (1973). Afriat and revealed preference theory. *Review of Economic Studies, 40*, 419–425.

Dobell, A. (1965). A comment on A. Y. C. Koo's an empirical test of revealed preference theory. *Econometrica, 33*, 451–455.

Donni, O. (2008). Household behavior and family economics. *The Encyclopedia of Life Support Systems* **Contribution 6.154.9**.

Gross, L., & Yellen, J. (2004). *Handbook of graph theory*. Boca Raton, FL: CRC Press.

Lundberg, S., & Pollak, R. (2007). Family decision-making. *The New Palgrave, dictionary of economics*, 2nd Edn. **forthcoming**.

Starr, R. (1969). Quasi-equilibria in markets with non-convex preferences. *Econometrica, 37*, 25–38.

Starret, D. (1972). Fundamental nonconvexities in the theory of externalities. *Journal of Economic Theory, 4*, 180–199.

Talla Nobibon, F., Hurkens, C., Leus, R., & Spieksma, F. (2009). Coloring graphs to avoid monochromatic cycles. *Manuscript*, University of Leuven.

Varian, H. (1982). The nonparametric approach to demand analysis. *Econometrica, 50*, 945–974.

Varian, H. (2006). Revealed preference. In M. Szenberg, L. Ramrattan, & A. A. Gottesman (Eds.) *Samuelsonian economics and the 21st century*. Oxford: University Press.