Heuristics for deciding collectively rational consumption behavior

Fabrice Talla Nobibon*, Laurens Cherchye†, Bram De Rock‡, Jeroen Sabbe§, Frits C.R. Spieksma¶

Abstract. We consider the computational problem of testing whether observed household consumption behavior satisfies the Collective Axiom of Revealed Preferences ($CARP$). We propose a graph such that the existence of a node-partitioning giving rise to two induced subgraphs that are acyclic implies that the data satisfy $CARP$. Furthermore, we propose and implement heuristics that are quite fast, that can be used to check reasonably large datasets for $CARP$ and that can be of particular interest when used prior to computationally demanding approaches. Finally, from the computational results we conclude that these heuristics can be effective in testing $CARP$.

Keywords: Collective model of household consumption; Collective Axiom of Revealed Preference; Pareto efficiency; Directed graph; Graph coloring; Graph partitioning; Acyclic subgraph; Heuristics.

---

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/decision makers) with their own rational preferences. It only assumes that household decisions are Pareto efficient, 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).

*University of Leuven, Operations Research Group, Naamsestraat 69, B-3000 Leuven, Belgium. E-mail: Fabrice.TallaNobibon@econ.kuleuven.be

†University of Leuven, Campus Kortrijk and Center for Economic Studies; Fund for Scientific Research - Flanders (FWO-Vlaanderen). E. Sabbelaan 53, B-8500 Kortrijk, Belgium. E-mail:laurens.cherchye@kuleuven-kortrijk.be.

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

§University of Leuven, Campus Kortrijk and Center for Economic Studies; Ph.D. Fellow of the Research Foundation Flanders (FWO-Vlaanderen). E. Sabbelaan 53, B-8500 Kortrijk, Belgium. E-mail:jeroen.sabbe@kuleuven-kortrijk.be

¶University of Leuven, Operations Research Group, Naamsestraat 69, B-3000 Leuven, Belgium. E-mail: Frits.Spieksma@econ.kuleuven.be

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). This 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, this can also be seen as a basic test for the whole class of such cooperative models. More generally, Pareto efficiency can be considered as a natural benchmark for analyzing the collective rationality of collective decisions, in cooperative as well as non-cooperative settings.

Cherchye, De Rock and Vermeulen (2007, 2008) introduced the Collective Axiom of Revealed Preferences ($CARP$) as a testable (nonparametric) condition for the general collective consumption model. More specifically, $CARP$ is a necessary and sufficient condition for observed household consumption behavior to be consistent with the collective consumption model. 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.

Recently, Cherchye et al. (2008) formulated the computational problem of verifying $CARP$ as an Integer Programming (IP) problem. They 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. Still it is well-known that solving IP problems with exact implicit enumeration methods is computationally demanding. In another study, Deb (2008a) proposes a heuristic for testing the collective model, yet he starts from a different condition which is sufficient but not necessary for $CARP$: data satisfying this condition satisfy $CARP$, but not necessarily vice versa. He shows that testing this condition is NP-complete.

In this paper we explore a graph-theoretical approach to deal with the computational problem of verifying whether observed household consumption behavior satisfies $CARP$. Facilitated by this graph-theoretical model, we propose heuristics to be able to quickly test 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 $CARP$. However, by performing computational experiments, we show that a vast majority of real-life instances is susceptible to our approach. This leads us to conclude that heuristics can be relevant for testing $CARP$, particularly for large datasets; see Cherchye et al. (2008) and Deb (2008a) for recent discussions of the relevance of testing $CARP$ for large instances. Moreover, not only can our heuristics serve as an alternative for exact and computationally demanding approaches like Integer Programming, our heuristics can also be used as a precursor before starting an exact algorithm; we refer to Section 5 for more details.

At a more general level, we demonstrate the usefulness of operations research techniques to implement nonparametric (revealed preference) conditions for economic decision behavior; our insights on testing $CARP$ consistency can also be instrumental for designing operational tests in alternative settings. For instance, 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, Ray and Snyder (2004).

The rest of the paper unfolds as follows. Section 2 defines collective rationality and the corresponding *CARP* condition. Section 3 introduces the graph formulation and establishes the computational complexity of the resulting problem. Section 4 presents the heuristics. Section 5 discusses the computational results. Section 6 concludes.

2. Collective rationality

Household consumption behavior that is consistent with the collective consumption model is said to be collectively rational. As indicated above, a collective rationalization of the data is possible if and only if the data are consistent with the *Collective Axiom of Revealed Preference* (*CARP*). This section provides formal definitions of the different concepts.

2.1 Collective rationalization

We consider a two-member household that purchases the (non-zero) N-vector of quantities $q \in \mathbb{R}^N_+$ with corresponding prices $p \in \mathbb{R}^N_{++}$. Generalizations for M-member households can be obtained along the lines of Cherchye, De Rock and Vermeulen (2007; supplemental material); this is also briefly discussed in Section 3. All goods can be consumed privately (e.g. member 1 uses the car alone), publicly (e.g. member 1 and 2 use the car together), or both. Generally, we have $q = q^1 + q^2 + q^h$ 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. 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. Note that this indeed implies that we only observe aggregate information, and do not have any information concerning the intra-household allocation. For ease of exposition, the scalar product $p_t' q_t$ is written as $p_t q_t$.

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. (Well-behaved means that the utility functions should satisfy 'local collective non-satiation'; this is the collective consumption analogue of standard local non-satiation concept for the individual consumption model. See Cherchye, De Rock and Vermeulen (2008) for more discussion.) Note that we do not demand that these utility functions are concave. (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 is problematic. See, for example, Starr (1969) and Starret (1972).)

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

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

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, in the following sense.

Definition 1 (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 $U^m(\widehat{q}_r) > U^m(\widehat{q}_t)$ implies $U^l(\widehat{q}_r) < U^l(\widehat{q}_t)$ $(m \neq l)$ for all feasible personalized quantities $\widehat{q}_r$ with $p_t q_r \leq p_t q_t$.*

2.2 Collective Axiom of Revealed Preference

This section defines *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, De Rock and Vermeulen (2007, 2008) for detailed discussions on *CARP* and the equivalence result.

Essentially, *CARP* imposes empirical restrictions on *hypothetical* member-specific preference relations H_0^m and H^m; these relations represent *feasible* specifications of the true individual preference relations that are consistent with the information that is 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 $u, \ldots, z \in \mathbb{T}$ with $q_s H_0^m q_u$, $q_u H_0^m q_v$,... and $q_z H_0^m q_t$. Thus given H_0^m for $m \in \{1, 2\}$, the transitive closure H^m follows. Note 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, De Rock and Vermeulen (2008), 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 1 if and only if the data is consistent with *CARP*. As such, we obtain the desired test of Pareto efficiency.

Definition 2 (CARP). *Let $S = \{(p_t, q_t); t \in \mathbb{T}\}$ be a set of observations. The set S satisfies the Collective Axiom of Revealed Preference (*CARP*) if there exist hypothetical relations H_0^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 $q_s H_0^1 q_t$ or $q_s H_0^2 q_t$;

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

$$\text{Rule 3: } \begin{cases} a) \text{ For } s,t \in \mathbb{T} : \text{if } p_t q_t > p_t q_s, \text{ then } \neg(q_s H^1 q_t) \text{ or } \neg(q_s H^2 q_t) \\ b) \text{ For } s, t_1, t_2 \in \mathbb{T} : \text{if } p_t q_t > p_t(q_{s_1} + q_{s_2}), \text{ then } \neg(q_{s_1} H^1 q_t) \text{ or } \neg(q_{s_2} H^2 q_t) \end{cases}.$$

Interestingly, this *CARP* condition 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 2 can again be interpreted in terms of Pareto efficiency. Specifically, Rule 2a states that, 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 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 expenditures $p_s q_s$) 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 l prefers q_s over q_{t_2}. Finally, Rule 3 complements Rule 2. Rule 3a states that, if q_s was attainable when q_t was chosen, then it cannot be that both members prefer q_s over q_t; otherwise Pareto improvements would have been possible (under the given prices p_t and outlay $p_t q_t$), which conflicts with collective rationality. Similarly, Rule 3b states that, if $q_{s_1} + q_{s_2}$ was attainable when q_t was chosen, then it cannot be that member 1 prefers q_{s_1} over q_t while, at the same time, member 2 prefers q_{s_2} over q_t.

3. A graph-theoretic formulation

Deciding whether the data S satisfy *CARP* is, in fact, a decision problem. In this section, we show how to build a directed graph $G(S) = (V(S), A(S))$ with the following property: if the nodes of $V(S)$ can be partitioned into two sets such that each induced subgraph is acyclic, then the data satisfy *CARP*. We also provide an example which shows that the converse is not necessarily true; that is, there exist instances for which the graph $G(S)$ does not admit a partition into two acyclic subgraphs while there exist H_0^1, H_0^2 satisfying Rules 1-3. Finally, we show that deciding whether such a partition into two acyclic subgraphs exists for our graph is NP-complete. In what follows we will, for reasons of notational convenience, simply write G, V, and A instead of $G(S), V(S)$, and $A(S)$ respectively. An equivalent way of phrasing the graph-theoretic problem is as follows: can we color each node of G red or blue such that no monochromatic cycle exists? (A monochromatic cycle is a collection of arcs $(v_1, v_2), (v_2, v_3), \ldots, (v_k, v_1)$ such that all v_i's have the same color). For an arbitrary directed graph G, the problem of node-partitioning the graph into two acyclic induced subgraphs was proven to be NP-complete by Deb (2008b). 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, Chen and Chen (2004) (who study the complexity of the problem for specific graph classes).

Let us now describe how the graph is built. Given a set of observations $S = \{(p_t, q_t); t \in \mathbb{T}\}$, each distinct pair of 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. 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 $G' = (V, A')$.

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 $u, v \in \mathbb{T}$ (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 $(t_2, u') = (v', s)$). Then 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 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 result shows that when the graph G can be node-partitioned into two acyclic subgraphs, the set of observations $S = \{(p_t, q_t); t \in \mathbb{T}\}$ satisfies $CARP$; that is there exist H_0^m, $m = 1, 2$ satisfying Rules 1-3. In other words, when we can color each node of the graph G with one of the two colors red or blue, such that $V = V_B \cup V_R$, $V_B \cap V_R = \emptyset$ and the induced subgraphs $G_B = (V_B, A_B)$, $G_R = (V_R, A_R)$ are each acyclic, the preference relations H_0^m, $m = 1, 2$ exist.

Theorem 1. *If the graph G can be node-partitioned into two acyclic subgraphs then the set of observations $S = \{(p_t, q_t); t \in \mathbb{T}\}$ satisfies* CARP*; that is there exists H_0^1, H_0^2 satisfying Rule 1-3.*

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.
Consider H_0^1 and H_0^2 defined by $q_s H_0^1 q_t$ if and only if $(s, t) \in V_B$ and $q_s H_0^1 q_s$ for all $s \in \mathbb{T}$. Similarly, $q_s H_0^2 q_t$ if and only if $(s, t) \in V_R$ and $q_s H_0^2 q_s$ 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) that is colored blue, we have $q_s H_0^1 q_t$. For each node (s, t) that 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 Rules 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 $q_t H^1 q_s$. $p_s q_s \geq p_s q_t$ 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). As $q_tH^1q_s$, 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. Since $G_B=(V_B,A_B)$ is an acyclic subgraph, $(s,t)\in V_R$ and hence $q_sH_0^2q_t$. Notice that a similar reasoning is applied to show that if $p_sq_s\geq p_sq_t$ and $q_tH^2q_s$ then $q_sH_0^1q_t$ for any observations s and t. This completes the proof that the **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 $p_sq_s<p_s(q_{t_1}+q_{t_2})$ 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 $p_sq_s\geq p_s(q_{t_1}+q_{t_2})$ and $q_{t_1}H^1q_s$. $p_sq_s\geq p_s(q_{t_1}+q_{t_2})$ implies that (s,t_1) and (s,t_2) belong to V. $q_{t_1}H^1q_s$ 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, $q_{t_1}H^1q_s$ 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 $q_sH_0^2q_{t_2}$. As in the proof of **Rule 2**: a), the symmetry between H_0^1 and H_0^2 allows this reasoning to be applied to show that if $p_sq_s\geq p_s(q_{t_1}+q_{t_2})$ and $q_{t_1}H^2q_s$, $q_sH_0^1q_{t_2}$ for any three distinct observations s,t_1,t_2. This completes the proof of the **Rule 2**: b).
Rule 3: a) As $V_B\cap V_R=\emptyset$ and $p_sq_s=p_sq_s$ 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 $p_sq_s>p_s(q_{t_1}+q_{t_2})$ and $q_{t_1}H^1q_s$ and $q_{t_2}H^2q_s$. $p_sq_s>p_s(q_{t_1}+q_{t_2})$ implies that $(s,t_1)\in V=V_B\cup V_R$. From $q_{t_2}H^2q_s$ and **Rule 2**: b), we know that $(s,t_1)\in V_B$. $q_{t_1}H^1q_s$ 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 satisfying Rules 1-3. □

Notice that the arguments used to prove Theorem 1 can be generalized for a household with $M\geq 2$ members. Hence, for a household with more than two members, if the graph G 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 Rules 1-3.

The following example shows how the graph is built from a specific set of observations using the procedure described above.

Example 1. *Consider a situation with* 3 *goods (*$N=3$*) and two household members (*$M=2$*), with the following three observed price-quantity combinations (*$T=3$*):*
$q_1=(8\quad 2\quad 2)'$; $q_2=(1\quad 8\quad 3)'$; $q_3=(1\quad 2\quad 8)'$; $p_1=(6\quad 2\quad 2)'$; $p_2=(2\quad 6\quad 1)'$; $p_3=(2\quad 3\quad 5)'$. *Notice that the following double sum inequalities hold:* $p_1q_1>p_1(q_2+q_3)$ *and* $p_2q_2>p_2(q_1+q_3)$. *The graph representation of this problem is given by Figure 1.*

In Figure 1, we have colored the nodes red and blue such that both subgraphs are acyclic. The result of Theorem 1 implies that the set of observations of Example 1 satisfies *CARP*.

Example 2 shows that the converse of Theorem 1 is not true.

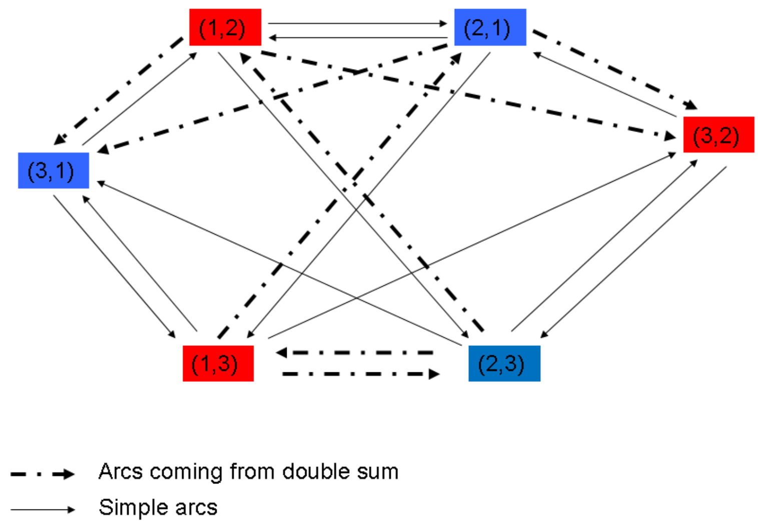

Figure 1: The graph built from the data of example 1.

Example 2. *Consider a situation with* 4 *goods* ($N = 4$) *and two household members* ($M = 2$)*, with the following four observed price-quantity combinations* ($T = 4$)*:*
$q_1 = (8 \quad 2 \quad 2 \quad 0)'$*;* $q_2 = (1 \quad 8 \quad 3 \quad 0)'$*;* $q_3 = (1 \quad 2 \quad 8 \quad 0)'$*;* $q_4 = (1 \quad 2 \quad 0 \quad 5)'$*;* $p_1 = (6 \quad 2 \quad 2 \quad 10)'$*;* $p_2 = (2 \quad 6 \quad 1 \quad 10)'$*;* $p_3 = (2 \quad 3 \quad 10 \quad 4)'$*;* $p_4 = (1 \quad 1 \quad 1 \quad 1)'$*. Notice that the following double sum inequalities hold:* $p_1q_1 > p_1(q_2 + q_3)$*,* $p_2q_2 > p_2(q_1 + q_3)$*,* $p_3q_3 > p_3(q_1 + q_4)$ *and* $p_3q_3 > p_3(q_2 + q_4)$*. The graph representation of this problem is given by Figure 2.*

In Figure 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.

However, it is easy to see that H_0^1 and H_0^2 defined as follows satisfy Rules 1-3. Define H_0^1 and H_0^2 by $q_1H_0^1q_2$, $q_1H_0^1q_3$, $q_3H_0^1q_2$, $q_3H_0^1q_4$ and $q_iH_0^1q_i$ for $i = 1, \ldots, 4$. $q_2H_0^2q_1$, $q_2H_0^2q_3$, $q_3H_0^2q_1$, $q_3H_0^2q_4$ and $q_iH_0^2q_i$ for $i = 1, \ldots, 4$. Notice that H_0^1 and H_0^2 have non-trivial intersection; that is there exist two distinct observations s, t such that $q_sH_0^1q_t$ and $q_sH_0^2q_t$. In fact, any H_0^1 and H_0^2 satisfying Rules 1-3 for this graph will have a non-trivial intersection. This non-trivial intersection of H_0^1 and H_0^2 is necessary for this example to hold. Therefore, if there exists H_0^1 and H_0^2 with only trivial intersection, then the corresponding graph can be partitioned into two acyclic subgraphs and the converse of Theorem 1 will hold.

Now, we show that deciding whether it is possible to partition the nodes of a graph $G = (V, A)$, which originates from the data of a collectively rational consumption behavior problem, into two sets such that each induced subgraph is acyclic, is NP-complete.

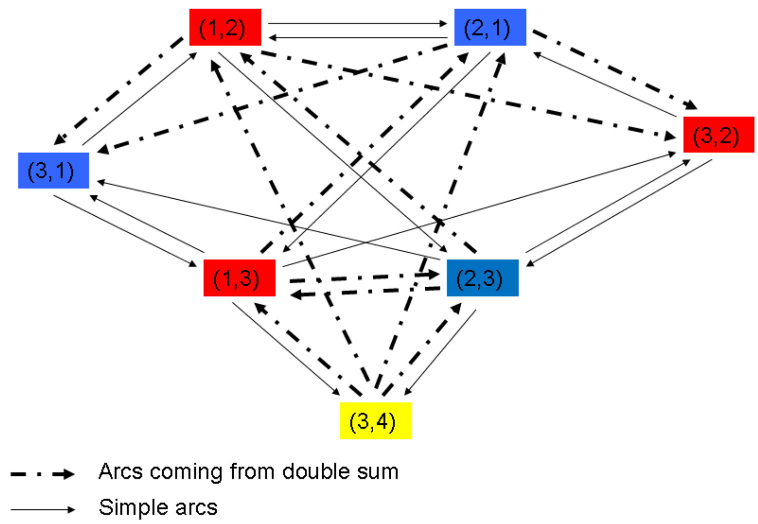

Figure 2: The graph built from the data of example 2.

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 2 acyclic subgraphs exists, is NP-complete.*

Proof: See the Appendix. □

Notice that this result does not imply that testing *CARP* is NP-complete; this is because Theorem 1 is not an equivalence. Theorem 2 shows that when one imposes that solutions should be found quickly (and we describe heuristics in the next section), the consequence is that there are instances allowing a feasible partition, which will not be found by the method employed (unless $P = NP$).

4. Heuristics

This section is devoted to simple heuristics for partitioning the graph $G = (V, A)$ described in Section 3. We first present an algorithm which partitions the graph G into two acyclic subgraphs when $G = G'$. Thus, we prove here that in case there are no double-sum arcs, the data satisfy *CARP*. 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.

4.1 The special case where $G = G'$

We present an algorithm which partitions the graph G into two acyclic subgraphs when $G = G'$. This corresponds to the case where there are no double sum arcs. Notice that the graph G still may contain a cycle.

Algorithm 1 Node-partitioning G when $G = G'$

for $t = 1, \dots, T-1$ **do**
    for $s = t+1, \dots, T$ **do**
        if $(t, s) \in V$ **then**
            color (t, s) red
        end if
        if $(s, t) \in V$ **then**
            color (s, t) blue
        end if
    end for
end for

The following result shows that Algorithm 1 partitions the graph G into two acyclic subgraphs when $G = G'$.

Lemma 1. *If $G = G'$, then Algorithm 1 partitions the graph G into two acyclic subgraphs.*

Proof: Applying Algorithm 1 yields a coloring of the nodes of G. Let $V_R = \{(s,t) \in V,\ (s,t)\,\text{red}\}$ and $V_B = \{(s,t) \in V,\ (s,t)\,\text{blue}\}$. Clearly, by construction, we have $V_R \cap V_B = \emptyset$ and $V_R \cup V_B = V$. It remains to show that the subgraph G_B induced by V_B (as well as the subgraph G_R induced by V_R) is acyclic. Since $G = G'$ there are no double sum arcs, hence, each arc goes from a node (s,t) to a node (t,u). Now, suppose that G_B is cyclic. Then there exists a sequence of distinct observations $t_1, t_2, \dots, t_n \in \mathbb{T}$ such that the nodes (t_i, t_{i+1}), $i = 1, \dots, n-1$ and (t_n, t_1) are in V_B (all these nodes are blue). However, $\exists i_0$ such that $t_{i_0} < t_{i_0+1}$ and $(t_{i_0}, t_{i_0+1}) \in V_B$ (otherwise, we have $t_1 > t_2 > \dots > t_n > t_1$, which is impossible). As $t_{i_0} < t_{i_0+1}$, Algorithm 1 colors (t_{i_0}, t_{i_0+1}) red and hence G_B is not cyclic, a contradiction. A similar argument shows that G_R is acyclic. □

Algorithm 1 runs in time $O(T^2)$. Moreover, it can be applied for any value of $M \geq 2$. In this case, at most two subgraphs are non-empty. Finally, we remark that this special case is quite relevant: we refer to Section 5 for more details.

4.2 Heuristics for arbitrary data

We distinguish coloring strategies on the one hand, and specific node orderings, or sequences, on the other hand. More specifically, we present 4 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.

4.2.1 Coloring strategies

CS1: This coloring strategy works as follows: 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: For a 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.

4.2.2 Ordering of the nodes

In the previous section, we assumed that a sequence of the nodes was given as input for each of the 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 $s<t$.

The next two sequences partition the nodes into those involved in a double-sum inequality, and those that are not. A node (s,t) is called *double-sum node* if there exist an observation u such that $p_s q_s \geq p_s(q_t+q_u)$ for some observations s and t. 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.

Sq5: Sequence 5 also uses the ordering of Sequence 1, but we place each double-sum node before each other node.

Sq6: Sequence 6 is the reverse of Sequence 5.

The following 6 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; 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.

Sq7: Sequence 7 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.

Sq8: Sequence 8 is the reverse of Sequence 7.

Sq9: Sequence 9 is found by sorting the nodes in increasing order of their indegree; if there is a tie we use the ordering of Sequence 1.

Sq10: Sequence 10 is the reverse of Sequence 9.

Sq11: Sequence 11 is found by sorting the nodes in increasing order of their outdegree; if there is a tie we use the ordering of Sequence 1.

Sq12: Sequence 12 is the reverse of Sequence 11.

Sq13: In this sequence, the position of a node is chosen randomly.

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. Indeed, we apply all these heuristics on the given instances, and we comment on their quality in Section 5.3.

5. Computational experiments

5.1 Data

Our goal is to investigate the usefulness of the graph construction from Section 3, and to assess the quality and the speed of the heuristics proposed above. 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 8 times. We refer to Cherchye et al. (2008) for more details on the data.

Data I consists of the same real-life instances as used by Cherchye et al. (2008); as such this allows us to compare the integer programming approach and the heuristics described here, see Section 5.3. In order to obtain bigger datasets 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 2 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 8 times, data set sizes are multiples of 8 and range from 8 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, Browning, and Crawford (2008) use, respectively, Spanish and Danish consumer panels with at most 24 observations per household.

5.2 Implementation

Building the set of nodes V

The data are the observations defined by $S = \{(p_t, q_t); t \in \mathbb{T}\}$ where $q_t \in \mathbb{R}^N_+$ are consumption bundles and $p_t \in \mathbb{R}^N_{++}$ corresponding prices ($t \in \mathbb{T} = \{1 \ldots, T\}$). From this data, we build the graph G as described in Section 3. Algorithm 2 depicts the steps to follow to derive the set V of nodes. It also identifies the nodes involved in the double sum inequalities. The time complexity of the Algorithm 2 is $O(T^3)$.

Algorithm 2 Build the set V of nodes

$V = \emptyset$ // set of nodes
$DS = \emptyset$ // nodes involved in the double sum inequalities
for $t = 1, \ldots, T-1$ **do**
  for $s = t+1, \ldots, T$ **do**
    if $p_s q_s \geq p_s q_t$ **then**
      $(s, t) \in V$
    end if
    if $p_t q_t \geq p_t q_s$ **then**
      $(t, s) \in V$
    end if
    for $t_2 = s+1, \ldots, T$ **do**
      if $p_s q_s \geq p_s(q_t + q_{t_2})$ **then**
        $(s; t, t_2) \in DS$
      end if
      if $p_t q_t \geq p_t(q_s + q_{t_2})$ **then**
        $(t; s, t_2) \in DS$
      end if
      if $p_{t_2} q_{t_2} \geq p_{t_2}(q_t + q_s)$ **then**
        $(t_2; t, s) \in DS$
      end if
    end for
  end for
end for

Building the set of arcs A

The arcs of the graph G' are easily identified. To build the arcs coming from the double sum, we proceed as follows. For a given node (s, t) involved in a double sum inequality (that is there exists t_1 such that $p_s q_s \geq p_s(q_t + q_{t_1})$), we use Dijkstra's algorithm (Ahuja, Magnanti and Orlin (1993)) to find all the nodes which are such that there is a path from (s, t) to those nodes. Among those nodes, we identify those ending with s (these are nodes $(., s)$) and draw an arc from (s, t_1) to the node $(t, .)$ appearing in each path.

Checking acyclicness of (V, A)

Clearly, in our heuristics we need to check often whether some induced subgraph is acyclic. We use the *topological ordering* algorithm, see Ahuja, Magnanti and Orlin (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. Its time complexity is $O(m)$ where m is the number of arcs.

We have implemented all algorithms in Visual Studio C++ 2005; all the experiments were run on a HP Pavilion dv6000 laptop with AMD Turion(tm) 64×2 Mobile Technology

TL-56 processor with 1.80 GHz clock speed and 2047 MB RAM, equipped with Windows Vista.

5.3 Computational results

Let us first consider the instances from 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.

Table 1, 2 and 3 give the properties of the graph representation of these instances. Notice that each graph contains at least one cycle; that is, each graph is cyclic. The analysis of these tables shows that 57 instances out of 69 can be partitioned into acyclic subgraphs using Algorithm 1 (see Subsection 4.2.1); that is because they have no double sum arc. This represents more than 82% of the instances! This clearly shows that it is worthwhile to detect the absence of double-sum arcs in the data; and if these arcs are absent one can use Algorithm 1 to get a conclusive answer whether the data satisfy *CARP* (instead of having to solve an IP-model).

We then apply the heuristics to the remaining 12 instances. Table 4 and 5 display the output of the heuristics. In each of these tables, a row (except for the first row and the last row) corresponds to a single instance. The column called "time" (which corresponds to a specific sequence) is expressed in seconds, and is the mean value of the time needed for the four strategies using that particular sequence. The column "Opt. CS" identifies the strategies for which we have obtained a partition into acyclic subgraphs. Finally, the last row gives, for each sequence, the number of strategies for which a feasible coloring was found.

From Tables 4 and 5, we see that for each instance except 1935-3-101, there is at least one heuristic finding a feasible coloring, meaning that each instance (except 1935-3-101) 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 Section 2 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 strategies 1 and 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 10 instances. This contrasts with strategies 2 and 3 which are only successful for 2 and 5 instances, respectively. Apparently, when coloring the nodes sequentially, it is better to keep using the same color, and only resort to another color when forced, 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 instance, there are often (but not always) many sequences for which this strategy is successful. Sequence 5 and 12 contain the highest number of strategies for which a feasible coloring was found, making them the most attractive sequences. In particular, the heuristic obtained by combining sequence 5 and strategy 1 (CS1) is very successful indeed: it solves all the instances except the one that is not solved by any heuristic (1935-3-101).

In fact, instance 1935-3-101 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 5) led to a partial feasible coloring of 4224 nodes, i.e., about 95% of the nodes; however, it still remains to be decided whether these data satisfy *CARP*.

Tables 4 and 5 also show that the heuristics are quite fast. Computing times for most instances are within 0.1 second, and always (except for 1935-3-101) within 2 seconds. This is in contrast with the computing times of Cherchye et al. (2008), who report computing times up to 5 minutes for their instances. It should be noted, though, that solving the IP-model leads 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, investing a little computation time to test for *CARP* quickly seems a sensible approach for Data I.

Now, we turn to the instances from Data II. The name of a group of instances is represented by "Rand" followed by a number. Each group contains 10 randomly generated instances. Rand is used to express the random characteristics of these instances and the number refers to the number of instances with 8 observations aggregated. For instance, Rand-5 has $8 \times 5 = 40$ observations as it is the aggregation of 5 instances, each with 8 observations.

Table 6 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 10 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 cycle and double sum arc. Therefore, instances with only cycle and no double sum arc are not counted.

Table 7 and 8 display the output of the heuristics when applied to the instances in Data II. The notations are the same as in Table 4 and 5; an entry in the column "Opt. 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 column time is the average over the 10 values obtained for the instances in that group. The last column of Table 8 (Nr. solved) reported the number of instance in each group for which the heuristics are able to find an optimal partition.

When analyzing the results of Table 7 and 8, we see that for the instances with less than or equal to 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 2 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 column of Table 8. Still, more than 60% of the instances whose number of observations is between 48 and 72 is 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. Notice that there are three possible situations: either a coloring exists, but the heuristics fail to find one, or the graph does not admit a coloring in spite of the fact that the data satisfy *CARP*, or the data simply do not satisfy *CARP*. More sophisticated heuristics might shed a light on this question.

Overall, Table 8 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 5 and 12 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

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

Instance	Nr observations	Nr nodes	Nr double sum	Nr simple arcs	Nr double sum arcs	Nodes involved in DS arcs	Total nr 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	7	22	2	52	1	2	53	11.47	1
1924-4-15	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	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	12	68	42	280	17	17	297	6.52	1
1927-5-27	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	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	16	118	30	682	17	15	699	5.06	1
1930-5-17	17	139	11	976	9	14	985	5.14	1

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

Instance	Nr observations	Nr nodes	Nr double sum	Nr simple arcs	Nr double sum arcs	Nodes involved in DS arcs	Total nr arcs	Density	Cyclic
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	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
1936-6-40	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	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

Table 2: Properties of the Graph representation of the instances of Data I (continued)

Instance	Nr observations	Nr nodes	Nr double sum	Nr simple arcs	Nr double sum arcs	Nodes involved in DS arcs	Total nr arcs	Density	Cyclic
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	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

Table 3: Properties of the Graph representation of the instances of Data I (continued)

Instances	Nr nodes	Sq1		Sq2		Sq3		Sq4		Sq5		Sq6	
		Time	Opt. CS	Time	Opt. CS	Time	Opt. CS	Time	Opt. CS	Time	Opt. CS	Time	Opt. CS
1924-3-7	22	0.00	1,2,3,4	0.00	1,3,4	0.00	1,2,3,4	0.00	1,4	0.00	1,2,3,4	0.00	1,2,3,4
1924-4-15	95	0.00	1,4	0.00	1,4	0.00	1,4	0.00	1,4	0.00	1,4	0.00	1,4
1926-4-11	48	0.00	1,4	0.00	1,2,4	0.00	1,3,4	0.00	1,4	0.00	1,3,4	0.00	1,4
1927-4-12	68	0.00	4	0.00	1,4	0.00	4	0.00	1	0.00	1,2,3	0.00	4
1927-5-27	279	0.02	-	0.09	-	0.01	-	0.09	-	0.09	1,4	0.02	-
1929-4-32	410	0.12	-	0.23	1	0.16	-	0.30	1	0.21	1	0.12	-
1930-4-16	118	0.00	1,4	0.00	1	0.00	1	0.00	1	0.00	1	0.00	1,4
1930-5-17	139	0.00	4	0.00	4	0.00	4	0.01	-	0.00	1,4	0.00	4
1935-3-101	4384	34.04	-	78.54	-	31.33	-	123.93	-	6.00	-	34.12	-
1936-6-40	755	0.73	4	1.10	-	0.70	-	1.35	-	1.54	1,4	0.73	4
1937-4-21	226	0.03	1	0.04	1	0.04	1,4	0.05	1,4	0.03	1	0.02	1
1941-3-26	294	0.08	1,4	0.01	1	0.01	1,4	0.01	1	0.07	1	0.08	1,4
×	×	×	16	×	14	×	16	×	12	×	22	×	16

Table 4: Output of heuristics for instances of Data I

Instances	Sq7		Sq8		Sq9		Sq10		Sq11		Sq12		Sq13	
	Time	Opt. CS	Time	Opt. CS	Time	Opt. CS	Time	Opt. CS	Time	Opt. CS	Time	Opt. CS	Time	Opt. CS
1924-3-7	0.00	1,4	0.00	1,2,3,4	0.00	1,4	0.00	1,2,3,4	0.00	1	0.00	1,2,3,4	0.00	1,4
1924-4-15	0.00	1	0.00	1	0.00	1	0.00	1,4	0.00	1	0.00	1,4	0.00	-
1926-4-11	0.00	1,4	0.00	1,2	0.00	1	0.00	1,2,3,4	0.00	1	0.00	1,2,3,4	0.00	1,2
1927-4-12	0.00	-	0.00	1,4	0.00	1,4	0.00	-	0.00	-	0.00	1,4	0.00	-
1927-5-27	0.14	-	0.05	1	0.06	-	0.04	-	0.09	-	0.08	1,4	0.03	-
1929-4-32	0.50	1	0.14	1	0.22	1	0.25	1,4	0.30	1	0.24	1	0.20	-
1930-4-16	0.01	-	0.00	1	0.00	-	0.00	1	0.01	-	0.00	1,3,4	0.00	-
1930-5-17	0.01	-	0.00	1	0.00	-	0.01	1,4	0.01	1	0.00	1	0.00	1
1935-3-101	788.55	-	22.54	-	27.77	-	6.09	-	363.63	-	17.65	-	12.25	-
1936-6-40	3.15	-	1.04	1	1.50	-	0.88	1	1.71	1	0.95	1	1.38	-
1937-4-21	0.06	-	0.00	-	0.05	-	0.02	1	0.04	-	0.01	-	0.01	-
1941-3-26	0.20	-	0.04	3	0.09	-	0.05	1	0.10	-	0.05	1	0.04	-
×	×	6	×	15	×	7	×	18	×	6	×	21	×	5

Table 5: Output of heuristics for instances of Data I (continued)

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

Table 6: Properties of the Graph representation of the instances of Data II

Instances	Nr nodes	Sq1		Sq2		Sq3		Sq4		Sq5		Sq6		Sq7	
		Time	Opt. CS	Time	Opt. CS	Time	Opt. CS	Time	Opt. CS	Time	Opt. CS	Time	Opt. CS	Time	Opt. CS
Rand-1	22	0.00	1,0,0,1	0.00	1,0,0,1	0.00	1,0,0,1	0.00	1,1,1,1	0.00	1,1,1,1	0.00	1,0,0,1	0.00	0,0,0,0
Rand-2	95	0.01	5,1,1,5	0.01	5,3,1,4	0.01	5,1,1,5	0.01	5,1,2,5	0.01	4,4,4,4	0.01	5,1,1,5	0.01	3,1,1,3
Rand-3	48	0.08	7,3,2,7	0.08	7,2,2,7	0.08	7,3,1,6	0.09	7,2,3,7	0.08	7,5,6,6	0.08	7,3,2,7	0.14	5,2,3,3
Rand-4	68	0.35	7,1,1,9	0.35	7,1,1,7	0.37	7,2,2,6	0.38	7,3,1,5	0.33	8,5,5,5	0.35	7,1,1,9	0.73	2,2,1,3
Rand-5	279	1.38	6,0,0,8	1.51	6,2,1,6	1.51	6,0,0,4	1.66	6,0,2,5	1.32	8,3,3,3	1.37	6,0,0,8	3.41	2,0,0,1
Rand-6	410	4.53	5,1,0,7	3.77	4,0,1,4	4.69	5,1,0,4	4.21	4,0,0,3	2.99	5,2,2,2	4.52	5,1,0,7	10.23	2,0,1,1
Rand-7	118	6.60	4,0,0,6	9.12	4,0,0,1	7.58	4,0,0,3	10.19	4,0,0,0	3.87	3,1,1,1	6.59	4,0,0,6	25.40	1,0,0,0
Rand-8	139	18.17	4,0,1,4	18.89	4,0,0,3	19.24	4,0,0,3	21.13	4,0,0,1	16.99	5,2,1,4	18.22	4,0,1,4	56.35	1,0,0,0
Rand-9	4384	19.97	1,0,0,3	25.58	0,0,0,2	24.21	1,0,0,1	31.06	0,0,0,2	33.88	5,2,1,3	19.97	1,0,0,3	100.94	0,0,0,0
Rand-10	755	34.51	0,0,0,2	38.11	0,0,0,0	42.03	0,0,0,1	46.70	0,0,0,0	10.54	0,0,1,2	34.43	0,0,0,2	164.76	0,0,0,0
Rand-11	226	38.98	0,0,0,1	52.29	0,0,0,0	58.57	0,0,0,0	88.99	0,0,0,0	21.67	1,0,0,0	38.95	0,0,0,1	347.73	0,0,0,0
Rand-12	294	56.26	0,0,0,1	80.11	0,0,0,0	92.67	0,0,0,0	136.29	0,0,0,0	0.80	0,0,0,0	56.47	0,0,0,1	537.32	0,0,0,0
×	×	×	105	×	87	×	85	×	82	×	131	×	105	×	38

Table 7: Output of heuristics for instances of Data II

Instances	Sq8		Sq9		Sq10		Sq11		Sq12		Sq13		Nr. solved
	Time	Opt. CS	Time	Opt. CS	Time	Opt. CS	Time	Opt. CS	Time	Opt. CS	Time	Opt. CS	
Rand-1	0.00	1,1,1,1	0.00	0,0,0,0	0.00	1,1,1,1	0.00	0,1,1,0	0.00	1,1,1,1	0.00	1,0,1,0	10
Rand-2	0.01	2,3,2,2	0.01	2,2,2,2	0.01	2,2,3,2	0.00	3,2,1,1	0.02	4,4,4,2	0.01	3,1,2,1	10
Rand-3	0.12	7,7,6,6	0.09	5,3,2,5	0.07	6,5,4,6	0.10	4,2,2,4	0.11	7,6,6,7	0.10	7,3,4,3	10
Rand-4	0.37	6,4,3,6	0.39	4,1,1,4	0.30	6,3,4,6	0.42	2,1,2,3	0.46	7,6,4,7	0.37	6,1,2,4	10
Rand-5	1.55	6,3,5,4	1.46	2,0,1,1	1.21	6,3,3,5	1.39	2,1,0,1	2.18	8,5,5,6	1.26	2,1,0,0	10
Rand-6	4.07	4,2,3,3	4.77	2,0,0,3	2.64	4,2,2,2	4.35	2,0,0,0	4.88	5,3,2,2	3.26	2,0,0,0	8
Rand-7	2.32	1,0,0,1	9.18	1,0,0,0	1.80	1,1,0,1	7.36	1,0,0,0	7.47	2,0,1,4	3.35	1,0,0,0	6
Rand-8	15.58	4,1,0,1	18.99	2,0,0,0	10.29	4,1,1,1	16.45	1,0,0,0	22.74	4,1,0,3	11.20	1,0,0,0	7
Rand-9	28.18	2,1,0,3	29.85	0,0,0,0	13.41	1,0,1,1	29.42	0,0,0,0	40.05	3,3,1,2	12.98	0,0,0,0	6
Rand-10	13.63	1,0,0,0	61.56	0,0,0,0	5.41	0,0,0,1	44.76	0,0,0,0	39.15	1,0,0,1	14.81	0,0,0,0	3
Rand-11	6.05	0,0,0,0	105.78	0,0,0,0	1.33	0,0,0,0	51.29	0,0,0,0	82.32	0,0,0,0	23.93	0,0,0,0	2
Rand-12	4.55	0,0,0,0	132.23	0,0,0,0	4.53	0,0,0,0	96.40	0,0,0,0	92.34	0,0,0,0	21.77	0,0,0,0	1
×	×	103	×	45	×	94	×	37	×	130	×	46	83

Table 8: Output of heuristics for instances of Data II (continued)

6. Summary and conclusions

We introduced a graph for addressing the computational problem of testing whether observed household consumption behavior satisfies the Collective Axiom of Revealed Preferences (*CARP*). More precisely, we obtained that the existence of a node-partitioning giving rise to two induced subgraphs that are acyclic implies that the data satisfies *CARP*. This graph representation allowed us to propose and implement heuristics that are quite fast and that can be used to check large datasets for *CARP*. Moreover, these can be used before using a computational demanding approach. Finally, our computational results suggest that these heuristics are very effective for testing *CARP*.

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 Section 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. $p_{x_k^l} q_{x_k^l} \geq p_{x_k^l}(q_{x_j^l} + q_{t^l})$ 5. $p_{t^l} q_{t^l} \geq p_{t^l}(q_{x_k^l} + q_{\bar{x}_j})$ 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. $p_{x_i^l} q_{x_i^l} \geq p_{x_i^l}(q_{x_k^l} + q_{u^l})$ 8. $p_{u^l} q_{u^l} \geq p_{u^l}(q_{x_i^l} + q_{\bar{x}_k})$ 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 Figure 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.

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

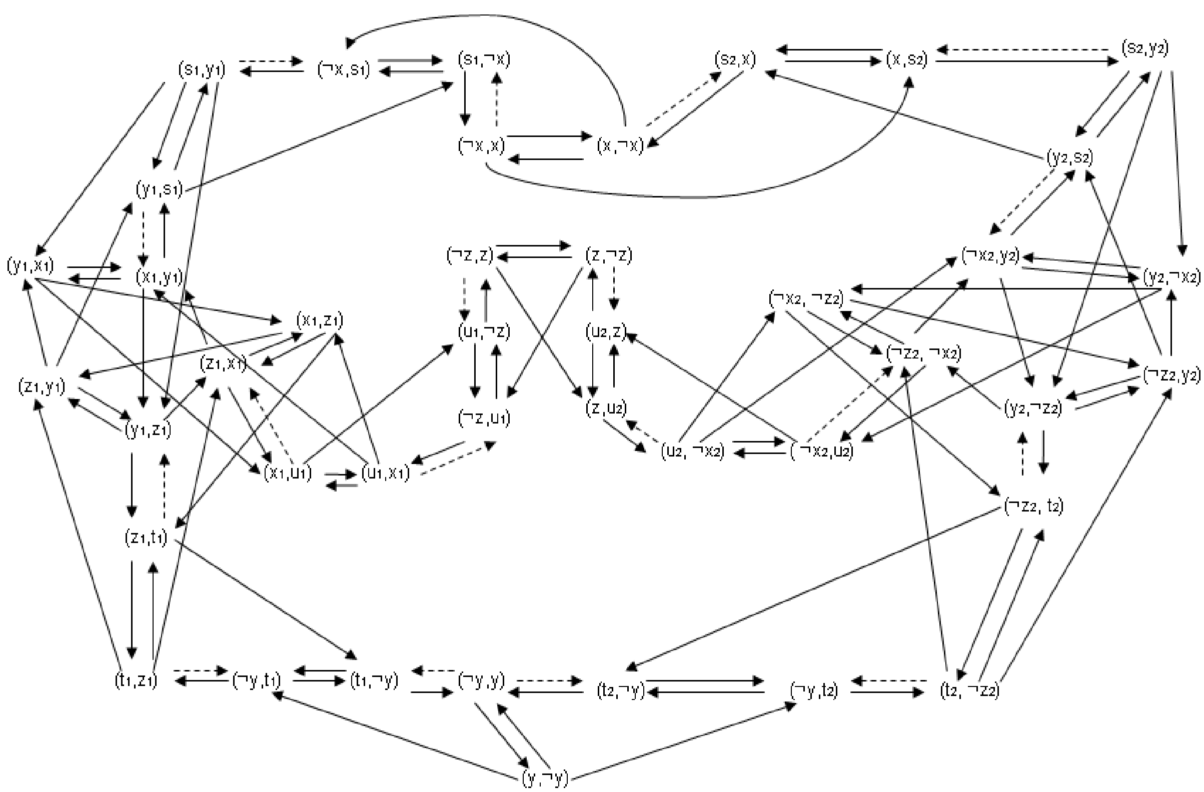

Figure 3: Example of reduction

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_j^l) 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_j^l) 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. $\square$

References

[1] Ahuja, R.K., T.L. Magnanti and J.B. Orlin. 1993. Network Flows Theory, Algorithms, and Applications. *Prentice Hall.*

[2] Blow, L., M. Browning and I. Crawford. 2008. Revealed preference analysis of characteristic models. *Review of Economic Studies.* **75** 371-389.

[3] Brown, D. and R. Matzkin. 1996. Testable restrictions on the equilibrium manifold. *Econometrica* **64** 1249-1262.

[4] Brown, D. and C. Shannon. 2000. Uniqueness, Stability and Comparative Statics in Rationalizable Walrasian Markets. *Econometrica* **68** 1529-1540.

[5] Browning, M. and P.-A. Chiappori. 1998. Efficient intra-household allocations: a general characterization and empirical tests. *Econometrica* **66** 1241-1278.

[6] Carvajal, A., I. Ray and S. Snyder. 2004. Equilibrium behavior in markets and games: testable restrictions and identification. *Journal of Mathematical Economics* **40** 1-40.

[7] Chang, G.J., C. Chen and Y. Chen. 2004. Vertex and tree arboricities of graphs. *Journal of Combinatorial Optimization* **8** 295-306.

[8] Chen, Z. 2000. Efficient algorithm for acyclic colorings of graphs. *Theoretical Computer Science* **230** 75-95.

[9] Cherchye, L., B. De Rock, J. Sabbe and F. Vermeulen. 2008. Nonparametric tests of collectively rational consumption behavior: an Integer Programming Procedure. *Journal of Econometrics*, forthcoming.

[10] Cherchye, L., B. De Rock and F. Vermeulen. 2007. The collective model of household consumption: a nonparametric characterization. *Econometrica* **75** 553-574.

[11] Cherchye, L., B. De Rock and F. Vermeulen. 2008. An Afriat theorem for the collective model of household consumption. *Working paper* University of Leuven, Belgium.

[12] Chiappori, P.-A. 1988. Rational household labor supply. *Econometrica* **56** 63-89.

[13] Chiappori, P.-A. 1992. Collective labor supply and welfare. *Journal of Political Economy* **100** 437-467.

[14] Christensen, M. 2007. Integrability of demand accounting for unobservable heterogeneity: a test on panel data. *IFS Working Paper W14/07* University of Manchester, UK.

[15] Deb, R. 2008a. An efficient nonparametric test of the collective household model. *Working paper* Yale University, United States.

[16] Deb, R. 2008b. Acyclic partitioning problem is NP-complete for K = 2. *Private communication* Yale University, United States.

[17] Donni, O. 2008. Household Behavior and Family Economics. *The Encyclopedia of Life Support Systems*, Contribution 6.154.9.

[18] Garey, M.R. and D.S. Johnson. 1979. Computers and intractability: A guide to the theory of NP-completeness. *W.H. Freeman and Co.*

[19] Lundberg, S. and R. Pollak. 2007. Family Decision-Making. *The New Palgrave, Dictionary of Economics, 2nd Edition*, forthcoming.

[20] Starr, R. 1969. Quasi-equilibria in markets with non-convex preferences. *Econometrica* **37** 25-38.

[21] Starret, D. 1972. Fundamental nonconvexities in the theory of externalities. *Journal of Economic Theory* **4** 180-199.