Journal of Artificial Intelligence Research 32 (2008) 1-36 Submitted 05/07; published 5/08

Enhancing Cooperative Search with Concurrent Interactions

Efrat Manisterski MANISTER@BIU.013.NET.IL
Department of Computer Science,
Bar-Ilan University, Ramat Gan, 52900 Israel

David Sarne SARNED@CS.BIU.AC.IL
Department of Computer Science,
Bar-Ilan University, Ramat Gan, 52900 Israel

Sarit Kraus SARIT@CS.BIU.AC.IL
Department of Computer Science,
Bar-Ilan University, Ramat Gan, 52900 Israel

Abstract

In this paper we show how taking advantage of autonomous agents' capability to maintain parallel interactions with others, and incorporating it into the cooperative economic search model results in a new search strategy which outperforms current strategies in use. As a framework for our analysis we use the electronic marketplace, where buyer agents have the incentive to search cooperatively. The new search technique is quite intuitive, however its analysis and the process of extracting the optimal search strategy are associated with several significant complexities. These difficulties are derived mainly from the unbounded search space and simultaneous dual affects of decisions taken along the search. We provide a comprehensive analysis of the model, highlighting, demonstrating and proving important characteristics of the optimal search strategy. Consequently, we manage to come up with an efficient modular algorithm for extracting the optimal cooperative search strategy for any given environment. A computational based comparative illustration of the system performance using the new search technique versus the traditional methods is given, emphasizing the main differences in the optimal strategy's structure and the advantage of using the proposed model.

1. Introduction

Coalition formation is well recognized as a key process in a multi-agent systems, mostly desirable in environments where a group of agents can perform a task more efficiently than any single agent can (Lermann & Shehory, 2000). In recent years many coalition formation models have been suggested, for various domains (Talukdar, Baerentzen, Gove, & de Souza, 1998; Dias, 2004), particularly for electronic commerce (Tsvetovat, Sycara, Chen, & Ying, 2000; Yamamoto & Sycara, 2001; Sarne & Kraus, 2005). In the latter context, the most common coalition is a coalition of buyers, derived mainly from the potential of obtaining volume discounts (Tsvetovat et al.; Sarne & Kraus, 2003) and the ability to search cooperatively for market opportunities in a more efficient manner (Sarne & Kraus, 2005).

©2008 AI Access Foundation. All rights reserved.

The cooperative economic search[1] incentive derives principally from the existence of search costs found in MAS. These costs reflect the resources (not necessarily monetary) that need to be invested or consumed while searching for opportunities in the environment (Sarne & Kraus, 2005) (e.g., searching for an opportunity to buy a product in the context of the electronic marketplace). The scenario of having search costs is common in MAS where the agent needs (for its decision making process) immediate information concerning market opportunities. Given the richness of opportunities and the dynamic and open nature of these environments, central mechanisms are usually incapable of supplying such information with the level of completeness and accuracy required by the agent, certainly not without a cost. Thus the agent needs to spend some of its resources on search related activities. Despite the reduction in the magnitude of these search costs in the electronic commerce era, the continuous growth in the number of retailers and virtual stores over the Internet, followed by a phenomenal increase in the number of opportunities available, makes the overall search cost an important parameter affecting buyers' search strategy (Choi & Liu, 2000; Kephart & Greenwald, 2002; Sarne & Kraus, 2005; Bakos, 1997).

In this context, the cooperative search offers the advantage of sharing, reusing and re-allocating opportunities among the coalition members (Sarne & Kraus, 2005). For example, using cooperative search agents can exploit opportunities which would have been discarded otherwise if each of the agents would have conducted an alternative separate search. Nevertheless, the process of forming and maintaining the coalition induces some overhead, derived mainly from the communication and coordination activities (Sarne & Kraus, 2003), thus the coalition should set its search strategy in a cost/effective manner.

A classic example of the above in traditional markets is the procurement management officer of a corporation. Instead of having each individual in the cooperation spend time and resources on locating its specific requested supplies, the task is delegated to the procurement management officer. Here, in addition to the price discounts obtained for aggregated demands of identical items, the procurement management officer becomes highly updated with the different offers and specific supplies available by the different merchants in the markets. As a result the cost of locating the best deal for each request becomes significantly smaller (in comparison to the equivalent search conducted by each of the individuals).

The basic concepts by which a coalition should manage the cooperative search, including an analysis and computational means for extracting its optimal search strategies, are given by Sarne and Kraus (2005). Nevertheless, the assumption used in that model for constructing the coalition's strategy is that the coalition interacts with one seller agent at a time. Such an assumption ignores the inherent strength of autonomous agents, which is their capability to efficiently interact with several other agents in parallel. This capability derives primarily from their improved communication capabilities and their ability to process an enormous amount of information in a short time compared to people. In this paper, we take advantage of this capability and incorporate it into the cooperative economic search model, supplying a comprehensive analysis of the resulting parallel cooperative search variant. As we show throughout the paper, the parallel model weakly dominates the existing sequential cooperative search model described by Sarne and Kraus: it has the potential of significantly improving the searchers' performance in various environments, and always guarantees reaching at least the performance of the existing cooperative search model. In particular, the parallel interaction is preferable whenever an agent's search cost is non-linear or combines fixed

1. As opposed to the classical AI search (Hart, Nilsson, & Raphael, 1968) in which an agent seeks a sequence of actions that will bring it from an initial state to a desired goal state.

components (e.g. operational costs), depending on the number of interactions maintained (e.g. advantage of size). In such cases, the adoption of the parallel technique by the coalition suggests a reduction in the average cost per interaction with seller agents.

Moreover, the improvement achieved using the parallel technique increases when there is a finite decision horizon (i.e., whenever the coalition has a deadline for finishing its search). In addition to the advantage of reducing the average cost per interaction, in finite horizon settings the coalition benefits from the fact that it can increase the intensity of the search and, thus, scan more opportunities (in comparison to the sequential search model described by Sarne & Kraus, 2005) prior to reaching the deadline.

While the integration of parallel interactions technique into a single agent's search process is quite intuitive, finding the optimal (overall utility maximization) strategy for the cooperative search case is not trivial at all. The major difficulty derives from the fact that different coalition members may have heterogeneous multi-attribute utility functions. Therefore, extracting the value encapsulated in future streams of opportunities is complex. We overcome this difficulty and present an algorithm for extracting the coalition strategy. Our algorithm facilitates the calculation process of the coalition strategy and it is polynomial in the number of parallel interactions. This is a significant improvement over the brute-force algorithm which is exponential in the number of parallel interactions.

Similar to the model introduced by Sarne and Kraus (2005), we apply the multi-attribute utility theory (MAUT) (Keeney & Raiffa, 1976) to analyze preferences with multiple attributes in our agent based search mechanism. This enables a set of preferences to be represented by a numerical utility function. We consider the agents to be heterogeneous, i.e. each having its own utility function. The model is general, though we emphasize several specific implementation aspects relating to the B2C market (businesses selling products or services to end-user consumers), where sellers can supply almost any demanded volume, and to the C2C market (transactions between consumers), where sellers offer single items for sale. Based on the proposed analysis, the coalition can calculate its optimal strategy given the utility functions of the coalition members and the specific environment in which it operates.

Three basic stages are common to all coalition formation models (Sandholm, Larson, Andersson, Shehory, & Tohme, 1999; Tsvetovat et al., 2000): coalition structure generation (where the agents form/join the coalition), executing the coalition task, and dividing the generated value among the coalition members. Among these three stages our focus is on finding the optimal search strategy for the coalition, given its structure and the opportunity distribution. As suggested by Sarne and Kraus (2005), the coalition operates in its environment alongside many other coalitions differing in their size, their members' utility functions and the products they are seeking. These other coalitions, as well as the different individual utility functions play an important role when studying the stability of a coalition and the issue of revealing the true utility function (truth telling). The analysis of these important issues is based on the ability to properly derive the coalition's utility given any specific self structure and the environment within which it operates.[2] This paper aims to supply this functionality, laying the foundation and enabling research of many of the important aspects of coalition formation given above in the context of cooperative search (truth telling, stability, payoff division, etc.).

2. The utility considered is the agents' reported, not necessarily true, utility function, since the goal is to extract the optimal strategy for a given input.

The main contributions of this work are fourfold: First, we formally model and analyze the parallel cooperative search problem of agents operating in a costly environment. The parallel cooperative search model is a general search model and can be applied in various domains in addition to the electronic marketplace that is used as a framework for our work. Second, we show that in many environments the parallel cooperative search outperforms the previous search strategies (either when each agent searches by itself or when using a cooperative sequential search). Furthermore, we draw attention to scenarios where sequential cooperative search is proven to be non-beneficial, however parallel cooperative search is a favorable technique. Third, we supply an algorithm that facilitates the calculation of the coalition's optimal strategy, and significantly reduces the complexities associated with the attempt to extract this strategy from an appropriate set of equations. Finally we provide a comprehensive analysis of the parallel model for a finite decision horizon. We draw attention to the significant improvement that can be achieved by integrating the parallel technique into cooperative search in the finite decision horizon.

The rest of the paper is organized as follows. Section 2 reviews the related work, emphasizing the uniqueness of the proposed parallel cooperative search model. The model description and its underlying assumptions are given in section 3. In section 4, we formally describe the performance of the coalition when using the parallel cooperative search as a function of the strategy used and present the complexities associated with extracting the optimal search strategy. By exploring the unique characteristics of the coalition's optimal strategy when using the cooperative parallel search we manage to overcome this computational complexity. This process is described in section 5. Consequently, we present an efficient algorithm for extracting the optimal cooperative search strategy. Some interesting properties of the new search model, with regard to the market in which it takes place, are illustrated in section 6. In section 7, a finite decision horizon variant of the model is discussed. The parallel cooperative search performance and its advantages over the current search models, both in infinite and finite decision horizons, are illustrated in section 8. Finally, we conclude with a discussion and suggested future research directions in section 9.

2. Related Work

In many scenarios autonomous agents in multi-agent environments may cooperate in order to perform tasks. The need for cooperation may arise either when an agent is incapable of completing a task by itself or when operating as a group can improve the overall performance (Breban & Vassileva, 2001; Lermann & Shehory, 2000; Tsvetovat et al., 2000). Group based cooperative behavior can be found in various domains, such as solving complex optimization problems (Talukdar et al., 1998), military and rescue domains (Dias, 2004), e-business applications (Tsvetovat et al., 2000; Yamamoto & Sycara, 2001) and many more. The recognition of the advantages encapsulated in teamwork and cooperative behaviors, is the main driving force of many coalition formation models in the area of cooperative game theory and MAS (Lermann & Shehory, 2000; Li, Rajan, Chawla, & Sycara, 2003; Shehory & Kraus, 1998). Many examples from the the extensive literature on coalition formation can be found in books and journals (Kahan & Rapoport, 1984). In the electronic market domain, most authors focus on coalitions formed to obtain volume discounts Tsvetovat et al., Yamamoto and Sycara. Additional coalition formation models for the electronic marketplace consider extensions of the transaction-oriented coalitions into long-term ones Breban and Vassileva, and for large-scale electronic markets Lermann and Shehory.

Traditionally, the majority of this research effort has focused on issues concerning the optimal division of agents into disjoint exhaustive coalitions (Sandholm et al., 1999; Yamamoto & Sycara, 2001), division of coalition payoffs Yamamoto and Sycara and enforcement methods for interaction protocols. Only a few authors have considered the coalition's problem of determining its strategy in the electronic commerce domain, once the coalition is formed (Ito, Ochi, & Shintani, 2002). Nevertheless, other than a single exception (Sarne & Kraus, 2005), none of the proposed models have considered a coalition search in a costly environment, and in particular none of them (including Sarne & Kraus, 2005) have made use of its capabilities to maintain parallel interactions.

The problem of a searcher operating in a costly environment, seeking to maximize his long term utility is addressed in classical search theory (Lippman & McCall, 1976; McMillan & Rothschild, 1994, and references therein). There are three main search models that can be found in the literature. The first search model is the Fixed Sample Size (FSS) model, introduced by Stigler (1961). In this model the searcher first chooses the sample size and then draws a single sample where all observations are made simultaneously. The second model is the Single Agent Sequential Search (SASS) strategy (Rothschild, 1974; Lippman and McCall). In this model the searcher draws exactly one observation at a time. Based on the value of the observations drawn till that time, the searcher decides whether to draw another observation. His decision depends upon what he observed. Attempts to adopt the sequential search model for agent-based electronic trading environments associated with search costs are suggested by Choi and Liu (2000), and Kephart and Greenwald (2002), though the main focus is on establishing the appropriate characteristics of the environment and search strategy rather than the computational aspects of extracting it. The last search method, Single Agent Parallel Search (SAPS) (Benhabib & Bull, 1983; Gal, Landsberger, & Levykson, 1981; Morgan, 1983; Morgan & Manning, 1985), encompasses the above search models as special cases. In this model the searcher may choose both the number of samples taken and the sample size in each period. This latter method, which outperforms the other two, is in fact the single agent's equivalent to our parallel cooperative search model considered in this paper. Nevertheless, search theory has mainly focused on a single searcher, looking for a single opportunity, either as a one sided (taking the environment's reaction to the search strategy used by the agent to be static) or two sided (as a matching model, analyzed from the equilibrium perspective) model. The analysis of a cooperative search is lacking. This, is in-spite of the fact that cooperative search has been proven (Sarne & Kraus, 2005) to be inherently different from a single agent's search in relation to its complexity, strategy structure and solution methodology.

Several extensions of economic search theory have been suggested for the case of a consumer searching for multiple different commodities, while facing imperfect information about prices (Gatti, 1999; Carlson & McAfee, 1984; Burdett & Malueg, 1981). Here, we can find different variants where the consumer visits one or more stores in order to minimize the total expenditure. Nevertheless, the attempt to adjust the proposed methods suggested in these models to support our parallel cooperative search process results in a solution complexity that is exponential in the number of parallel interactions. In contrast our algorithm for extracting the optimal search strategy is polynomial in the number of parallel interactions.

3. The Parallel Cooperative Search Model

We base our model description and formulation on the definitions given by Sarne and Kraus (2005) and extend them to reflect the agents' parallel search capabilities. We consider an electronic marketplace where numerous buyer and seller agents can be found. Each agent is interested in buying or offering to sell a well defined product. A product can be offered by many different seller agents under various terms and policies (including price). We assume that while buyer agents are ignorant of individual seller agents' offers, they are acquainted with the overall distribution of opportunities (where an opportunity is defined as the option to buy the product under specific terms and policies) in the marketplace.

Assuming there are no central mechanisms or mediators which can supply the agents with full immediate information concerning current market opportunities, each agent needs to search for appropriate opportunities to buy its requested product. Throughout the search the buyer agents locate seller agents and learn about their offers by interacting with them. Each buyer agent evaluates opportunities using its own multi-attribute utility function. Buyer agents may have heterogeneous preferences and thus the utility from a given opportunity differs according to the evaluating buyer agent.

In its most basic form, each buyer agent interacts with several sellers in parallel at each stage of its search thus learns about a new set of opportunities. Based on the agent's evaluation of the utility that can be gained from each opportunity in the set, the agent makes a decision whether to exploit any of the opportunities it encountered throughout its search (i.e. buy from any of the sellers) or resume its search in a similar method. A decision to resume the search is always accompanied by the number of parallel interactions to be conducted next.

The search activity is assumed to be costly (Choi & Liu, 2000; Kephart & Greenwald, 2002; Sarne & Kraus, 2005; Bakos, 1997). For each search stage in which the buyer agent locates, interacts and evaluates seller agents, the process induces a search cost. This cost is a function of the number of parallel interactions initiated and maintained by the agent. The search cost structure is principally a parameter of the market's liquidity and volatility, and thus it is assumed to be shared by all buyer agents operating in the specific marketplace. Recognizing the benefits of a cooperative search, buyer agents, interested in similar products or interchangeable products, may form coalitions (Sarne & Kraus, 2005). There are various methods by which the coalition members can coordinate their cooperative search (e.g. assign a representative agent to search on behalf of the coalition or simply take turns searching), each deriving a different search cost overhead structure. The coalition's search costs are assumed to increase both as a function of the number of parallel interactions it forms and the number of buyer agents in the coalition.[3] We assume a buyer agent's utility from a given opportunity may be interpreted into monetary terms. Thus the utilities are additive and the total search utility can be obtained by subtracting the search cost that the process induces from this value.

As part of its search process, the coalition needs to set a strategy for determining, given any set of known opportunities, whether to terminate or resume its search. In the latter case, the coalition also needs to determine the number of parallel interactions to be used in the next search round. The optimal strategy is the one maximizing its expected total search utility (opportunities utility minus search costs). As discussed in detail in the cooperative search model (Sarne & Kraus, 2005), given the option of side-payments the overall utility maximization strategy taken by the coalition is always the preferred one by all coalition members (i.e. no conflict of interests), regardless of the pre-set coalition's payoff division protocol. Given the coalition's goal of maximizing the overall coalition utility, its decision is not influenced by the payoff division protocol, nor by coalition

3. The reason for correlating the coalition's search cost with the number of coalition members is mainly associated with some coordination overhead (Sarne & Kraus, 2005).

stability considerations, but rather influences these two factors (Sarne & Kraus, 2005). Any of the agents' pre-determined portion of the coalition's utility will increase in its absolute value along with the increase of the net coalition utility, thus the overall utility maximization strategy is the preferred strategy by all agents at every stage of the search.

4. Parallel Cooperative Search (PCS) Analysis

The following section formally defines the search environment and the coalition's search strategy. For convenience, all the notations given, and their meanings, are summarized in a table at the end of the paper.

Let $B = (B_1, B_2, ..., B_k)$ be the set of the attributes defining any of the potentially available opportunities in the market, where each attribute B_i can be assigned a value from the finite set $(b^i_{min}, ..., b^i_{max})$. An opportunity's type is defined by the vector $\vec{o}_i = (b_1, b_2, ...b_k)$, assigning a value b_i to each specific attribute B_i.[4] We use O_p to denote the space of potential opportunity types the coalition may encounter. The opportunity types' distribution in the marketplace is denoted by the probability function $p(\vec{o})$, $\sum_{\vec{o} \in O_p} p(\vec{o}) = 1$. We consider a coalition $A = \{a_1, a_2, ..., a_{|A|}\}$ of a general size, where a_j is the $j-th$ buyer agent in the coalition. Each buyer agent, a_j, evaluates opportunities using a utility function $U_j : O_p \rightarrow R$, where $U_j(\vec{o})$ is the agent's utility from opportunity type $\vec{o}$. We denote the search cost associated with having a coalition of n agents maintaining w simultaneous interactions with seller agents over a search round by the function $c(w, n)$.

Let Θ be the collection of all possible sets of opportunities in the environment in which the agents reside. Given a set of known opportunities $\theta_{known} \in \Theta$ the coalition needs to determine its strategy (whether to terminate/resume the search and the number of parallel interactions in the later case). Similar to the analysis suggested for the Sequential Cooperative Search (SCS) model (Sarne & Kraus, 2005) we can reduce the large number of world states in which a coalition can be, by adopting a representation of states through sets of effective known opportunities. For that purpose we consider a function $alloc : \Theta \rightarrow O_p^n$ that maps a given set of opportunities θ to the coalition members in A (i.e. an allocation) in a way that the aggregated agents' utility when using this allocation is maximized.[5] In B2C markets the same opportunity may be allocated to more than one agent, while in C2C markets each opportunity is restricted to only one agent. Given a coalition A, we use $alloc(\theta) = (\vec{y_1}, ...\vec{y_n})$, $\vec{y_i} \in (\theta \cup \{\emptyset\})$ to represent the allocation resulting from applying the function $alloc$ over the set θ, where $\vec{y_i}$ denotes the opportunity associated with agent a_i ($y_i = \emptyset$ denotes that no opportunity was allocated to agent a_i).

The computation method used by the function $alloc$ is market-dependent. While in B2C markets the function assigns each agent the opportunity that maximizes its utility, $\vec{y_j}^* = \arg\max_{\vec{y} \in \theta} U_j(\vec{y}), j = 1, ..., n$, in C2C markets $alloc$ can be computed by solving a maximum weighted matching in a bipartite graph (Avis & Lai, 1988). Specifically for a set of opportunities $\theta \in \Theta$ found in the C2C market we construct a graph $G_\theta = (V_1, V_2, E)$, where each vertex of V_1 corresponds to an agent in A and each vertex in V_2 corresponds to an opportunity $\vec{o} \in \theta$. An edge connects an agent a_j in V_1 and an opportunity $\vec{o}$ in V_2 (each member in the two groups has edges connecting it to all the members

4. Notice that $\vec{o}$ is noted as a vector since it assigns a specific value to each of the different attributes, terms and conditions associated with a specific opportunity. For example, a specific opportunity to buy a calculator can be represented by the vector $\vec{o} = (scientific, 20\$, small\ display, pocket, 1YR\ warranty)$.
5. If there is more than one allocation that maximizes the overall coalition utility then the function $alloc$ chooses one of them according to a pre-defined order.

of the other group). The weight of such an edge is the utility for agent a_j from opportunity $\vec{o}$, $U_j(\vec{o})$. Here $alloc(\theta) = (\vec{y_1},..,\vec{y_n})$, where $\{(a_1,\vec{y_1}),..,(a_n,\vec{y_n})\}$ is a maximum weighted matching in G_θ. To illustrate the computation used by the function $alloc$, we use the following example:

Example 1. *Consider the following environment:*

Environment 1. *There are three agents, a_1, a_2 and a_3, searching for a product (e.g., memory chip) characterized by two attributes, B_1 (e.g., quality) and B_2 (e.g., store rating). Each attribute can have either the value 1 or 2, with an equal probability of 1/2. This means that there are four possible opportunities $\vec{o}_1 = (1,1)$ (both attributes' values are equal to 1), $\vec{o}_2 = (1,2)$, $\vec{o}_3 = (2,1)$, and $\vec{o}_4 = (2,2)$. The utility function of agents a_1, a_2 and a_3 are $U_1(\vec{o}) = 9B_1 + B_2$, $U_2(\vec{o}) = 4B_1 + 5B_2$, and $U_3(\vec{o}) = B_1 + 10B_2$, respectively. Table 1 summarizes the environment's setting.*

Opportunity	*(Attribute1, Attribute2)*	*Probability*	*Utility*		
			Agent a_1	*Agent a_2*	*Agent a_3*
$\vec{o}_1$	(1,1)	$\frac{1}{4}$	10	9	11
$\vec{o}_2$	(1,2)	$\frac{1}{4}$	11	14	21
$\vec{o}_3$	(2,1)	$\frac{1}{4}$	19	13	12
$\vec{o}_4$	(2,2)	$\frac{1}{4}$	20	18	22

Table 1: Agents' utilities for the four opportunities in Environment 1

Assume the coalition has already interacted with 4 sellers, encountering two opportunities of type $\vec{o}_1$ and two single opportunities of types $\vec{o}_3$ and $\vec{o}_4$. Here, the set of known opportunities is $\theta_{known} = \{\vec{o}_1,\vec{o}_1,\vec{o}_3,\vec{o}_4\}$.

We first calculate $alloc(\theta_{known})$ for the coalition operating in the B2C market. In this market we assume that sellers can supply any demanded volume. Therefore the allocation that maximizes the coalition's overall utility is assigning each agent the opportunity that maximizes its utility over θ_{known}. Since opportunity $\vec{o}_4$ maximizes the utility of each of the agents, we obtain $alloc(\theta_{known}) = (\vec{o}_4,\vec{o}_4,\vec{o}_4)$.

In C2C markets the above allocation is impossible, since opportunity $\vec{o}_4$ can be assigned only to one of the agents (each seller offers a single item for sale). In this case the optimal allocation can be calculated by solving a maximum matching problem resulting in the assignment $alloc(\theta_{known}) = (\vec{o}_3,\vec{o}_1,\vec{o}_4)$.

Given a set of known opportunities θ_{known}, we can use the function $alloc$ and its consequent allocation $alloc(\theta_{known})$ to calculate the immediate utility of the coalition if it terminates the search at the current time point. This utility, defined as the aggregated agents' utility when each of the agents is allocated according to allocation $alloc(\theta_{known})$, is denoted by $V_t(\theta_{known})$ (abbreviation for $V_{terminate}$) and can be calculated using:

$$V_t(\theta_{known}) = \sum_{j=1}^{n} U_j(\vec{y_j}) \tag{1}$$

where $alloc(\theta_{known}) = (\vec{y_1},...\vec{y_n})$ and $U_j(\emptyset) = 0, \forall j$.

Notice that up to this point, the world state space upon which the coalition defines its strategy is defined by the set of opportunities θ_{known} known to the coalition. This space is potentially very large. In order to reduce the strategy's space, we introduce the concept of equivalence between different sets of opportunities within the context of cooperative search. We consider two sets of opportunities $\theta', \theta'' \in \Theta$ to be equivalent ($\theta' \equiv \theta''$), if the following hold: (a) $V_t(\theta'') = V_t(\theta')$; and (b) $V_t(\theta' \cup \theta) = V_t(\theta'' \cup \theta)$ for any set of opportunities $\theta \in \Theta$ that the coalition may encounter in the future. For any two equivalent sets θ', θ'', the coalition is indifferent to knowing the opportunities in θ' or the opportunities in θ''. This is because any set of opportunities the coalition will encounter in the future results in a similar utility, thus the coalition's overall utility will be the same in both cases. Moreover, since the coalition's decisions are merely determined by its overall utility and in both cases similar utilities are reached with similar probabilities, the coalition uses the same search strategy (either terminates the search or uses the same number of parallel interactions in the subsequent search stage) for both opportunities sets.

Notice that according to the definition above *equivalent* is a transitive relation ($\theta' \equiv \theta''$, $\theta'' \equiv \theta''' \rightarrow \theta' \equiv \theta'''$). Moreover, $\theta' \equiv \theta''$ implies that $(\theta' \cup \theta) \equiv (\theta'' \cup \theta)$, $\forall \theta \in \Theta$. Given an allocation $\ell = (\vec{y}_1, ..., \vec{y}_n)$ of a set θ, we use $\{\ell\}$ to denote the set of opportunities that appear in ℓ.

Similar to former cooperative search models (Sarne & Kraus, 2005; Manisterski, 2007) the following theorem holds in our cooperative search model (and can be proven in a similar manner).

Theorem 1. *Any set of opportunities θ is equivalent to the set of opportunities returned by the function $alloc(\theta)$. Formally stated: $\theta \equiv \{alloc(\theta)\}$.*

Theorem 1 enables us to reduce the set of known opportunities that affect the coalition's strategy. The immediate implication of the Theorem 1 is that the coalition's strategy is affected only by the subset of θ_{known} defined by $\{alloc(\theta_{known})\}$. Thus the coalition does not need to keep all its known opportunities. It can reduce its set of known opportunities with which it determines its strategy to a subset, s. Given this result, we define *state* to be the set s of opportunities that are members of $\{alloc(\theta_{known})\}$. Formally, we can calculate the state s of a coalition A acquainted with a set θ_{known} of known opportunities by using the function $s = state(\theta_{known}) = \{alloc(\theta_{known})\}$. We use S_A to denote the set of all possible states of a coalition A. This definition significantly simplifies our analysis and enables the coalition to calculate its optimal strategy exclusively based on the sets of opportunities that are in S_A. The following example illustrates the computation of a state.

Example 2. *Consider the environment and the set of known opportunities described in Example 1. As computed in Example 1, the allocation that maximizes the coalition's overall utility in the B2C market is $alloc(\theta_{known}) = (\vec{o}_4, \vec{o}_4, \vec{o}_4)$. Thus the coalition's current state in the B2C market is $state(\theta_{known}) = \{\vec{o}_4\}$. As a result the coalition can ignore the other opportunities it encountered $\{\vec{o}_1, \vec{o}_1, \vec{o}_3\}$ and its strategy (terminate or resume the search and the number of parallel interactions in the later case) is the same as its strategy when $\theta_{known} = \{\vec{o}_4\}$. Similarly, the coalition's state in the C2C market includes the opportunities in $alloc(\theta_{known})$ thus $state(\theta_{known}) = \{\vec{o}_3, \vec{o}_1, \vec{o}_4\}$.*

The coalition's transition from one state to another during the search in the B2C and the C2C markets can be described as a directed acyclic graph (DAG). The vertices of this graph present all potential coalition states. A directed edge (s, s') connects two states, s and s', if there is an opportunity $\vec{o} \in O_p$ that changes the current coalition's state from s to s' (i.e $\exists \vec{o}$, s.t $state(s \cup \vec{o}) = s'$). To better understand the use of the DAG in these markets, we use the following two environments:

Environment 2. *There are two agents a_1 and a_2 searching for a product (e.g., a computer mouse) in a B2C market associated with 3 types of opportunities (e.g., 3 models): $\vec{o}_1$, $\vec{o}_2$ and $\vec{o}_3$. Table 2 summarizes the environment's setting.*

Opportunity	*Utility*	
	Agent a_1	*Agent a_2*
$\vec{o}_1$	5	10
$\vec{o}_2$	10	5
$\vec{o}_3$	20	21

Table 2: Agents' utilities for the three opportunities in Environment 2

Environment 3. *There are two agents a_1 and a_2 searching for a product (e.g., a used book) in a C2C market associated with two types of opportunities (e.g., English edition and American edition): $\vec{o}_1$ and $\vec{o}_2$. Table 3 summarizes the environment's setting.*

Opportunity	*Utility*	
	Agent a_1	*Agent a_2*
$\vec{o}_1$	5	10
$\vec{o}_2$	10	5

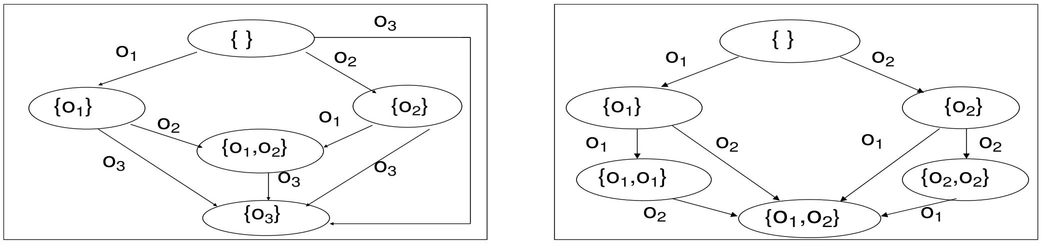
Table 3: Agents' utilities for the two opportunities in Environment 3

Figure 1(a) and Figure 1(b) show the DAG of states for Environment 2 (the B2C market) and Environment 3 (the C2C market) described below, respectively. Notice that it is possible to have opportunities that do not change the coalition's current state (since these opportunities do not increase the coalition's overall utility). To simplify the graph we did not mark these opportunities. It is notable that in the C2C market all sets that include up to n opportunities are feasible states (state s is feasible if it belongs to S_A, i.e. there is a set θ such that $state(\theta) = s$). This does not hold in the B2C market. For example, the set $\{\vec{o}_1, \vec{o}_3\}$ is not a feasible state since $\vec{o}_3$ maximizes all agents' utility. Thus if the coalition encounters this opportunity all other opportunities can be ignored.

Any coalition A that reaches a state s along its search can change its state to s' only if there is a sequence of directed edges from state s to state s'. The coalition can conduct parallel interactions, thus can transition within a single search round to a state s' which is not directly connected to state s. For example, in Environment 3, when in state $s = \{\}$ the coalition can change its state to $s' = \{\vec{o}_1, \vec{o}_2\}$ after one search round (even though there is no a directed edge between them). This can happen when the coalition conducts two or more parallel interactions in which it encounters opportunities $\vec{o}_1$ and $\vec{o}_2$. The transition to a new state suggests that in the new state the coalition has a termination utility equal to or higher than the utility at its current state.

We define a strategy as a function $x : S_A \rightarrow N$, where $x(s) = 0$ if the agent decides to terminate its search; otherwise $x(s)$ is the number of parallel interactions the coalition should maintain next, when in state s. We denote the optimal strategy by x^*.

We define $V(s,w)$ as the coalition's expected utility when using w parallel interactions when in state s (assuming any search decision taken at a future state $s' \neq s$ will make use of the optimal number of parallel interactions). The term $V(s,0)$ denotes the immediate utility obtained, if the



(a) (b)

Figure 1: States Diagram of (a) Environment 2 (B2C market); (b) Environment 3 (C2C market).

coalition decides to terminate the search at state s, thus: $V(s,0) = V_t(s)$. The value w ($w \in N$, $w \geq 0$) that maximizes the coalition's expected utility $V(s,w)$, is equal to $x^*(s)$:

$$x^*(s) = \arg\max_{w} V(s,w) \tag{2}$$

The coalition's expected utility from this point onwards when using its optimal strategy, denoted $V^*(s)$, can be expressed as:

$$V^*(s) = max_w V(s,w) = V(s,x^*(s)) \tag{3}$$

In order to formulate the appropriate equation for $V(s,w)$ (from which $x^*(s)$ and $V^*(s)$ can be derived) we make use of several additional notations and definitions. Consider a search round in which the coalition interacts simultaneously with w seller agents, yielding a set $\theta_w = \{\vec{o}_1, ..., \vec{o}_w\}$, $\vec{o}_i \in O_p$ of opportunities. Let Θ_w be the collection of all w-sized sets of opportunities that can be produced in the environment in which the coalition operates. We denote by $p_w(\theta_w)$ the probability of encountering a specific set of opportunities θ_w, when maintaining w random interactions with seller agents.

Similar to the basic cooperative search (Sarne & Kraus, 2005) we divide the w-sized opportunities space, Θ_w, into two sub-spaces, containing improving and non-improving w-sized sets of opportunities for the coalition's utility, respectively. Nevertheless, this definition needs to be extended to fit the scenario of parallel search as follows. Given the number of simultaneous interactions, w, and a state s, let $\Theta^s_{improve_w}$ be the collection of all w-size sets of opportunities, θ_w, that change the coalition's current state (formally stated as: $\Theta^s_{improve_w} = \{\theta_w | \theta_w \in \Theta_w \textit{ and } state(s \cup \theta_w) \neq s\}$). We denote the complementary set of $\Theta^s_{improve_w}$ by $\Theta^s_{stay_w}$ (the set that includes all w-size sets of opportunities θ_w that do not change the coalition's current state).

Therefore when the coalition encounters the set of opportunities θ_w, we can distinguish two possible scenarios:
(1) θ_w belongs to $\Theta^s_{stay_w}$ and the coalition's current state is still s. In this case the coalition's future expected utility (i.e., from this point on) remains $V(s,w)$. This is derived from the stationary nature[6]

6. Stationary strategy is a strategy in which each player chooses the same moves in every structurally equivalent subgame (Baron & Ferejohn, 1989).

of the problem - if no better state has been reached, the search resumes using the same number of parallel interactions w, yielding the same expected utility.
(2) θ_w belongs to $\Theta^s_{improve_w}$ and the coalition's current state changes to $s' = state(s \cup \theta_w) \neq s$. Since we assume that the coalition's decision taken at a future state $s' \neq s$ will make use of the coalition's optimal strategy $x^*(s')$, the coalition's expected utility can be expressed as $V^*(state(s \cup \theta_w))$.

By using the above analysis, the expected utility when using w parallel interactions while in state s, $V(s,w)$, can now be expressed as ($\forall w > 0$):

$$V(s,w) = -c(w,n) + \sum_{\theta_w \in \Theta^s_{improve_w}} p_w(\theta_w) V^*(s') + \sum_{\theta_w \in \Theta^s_{stay_w}} p_w(\theta_w) V(s,w) \tag{4}$$

where s' is the coalition's new state after it encounters a set of opportunities θ_w, $s' = state(s \cup \theta_w)$. After applying some basic mathematical manipulations on the above term we obtain:

$$V(s,w) = \frac{-c(w,n) + \sum_{\theta_w \in \Theta^s_{improve_w}} p_w(\theta_w) V^*(s')}{1 - \sum_{\theta_w \in \Theta^s_{stay_w}} p_w(\theta_w)} \tag{5}$$

Since $1 - \sum_{\theta_w \in \Theta^s_{stay_w}} p_w(\theta_w) = \sum_{\theta_w \in \Theta^s_{improve_w}} p_w(\theta_w)$ we obtain:

$$V(s,w) = \frac{-c(w,n) + \sum_{\theta_w \in \Theta^s_{improve_w}} p_w(\theta_w) V^*(s')}{\sum_{\theta_w \in \Theta^s_{improve_w}} p_w(\theta_w)} \tag{6}$$

Notice that in the case where no new better state can be reached, the denominator becomes zero, and $V(s,w) = -\infty$. This is quite straightforward since the coalition maintains an endless costly search (trying to reach a better state that actually does not exist). Here, inevitably the coalition's optimal strategy is to terminate the search. This important characteristic is used later to design the proposed algorithms for extracting the optimal search strategy.

At this point, one may attempt to compute the coalition's strategy x^* by solving a set of equations of types 1, 3 and 6. Nevertheless, this straightforward solution approach is accompanied by many inherent complexities, derived from the structure of the equations. First, notice that Equation 6 is a recursive equation and one needs to know the optimal strategy taken in future states s' when extracting the optimal strategy of a given state s. Second, the computation of $V(s,w)$ in Equation 6 is exponential in the number of parallel interactions, w, used (affecting the number of sets in $\Theta^s_{improve_w}$, both in the denominator and numerator). Last, according to the above formulation, the potential number of parallel interactions that may be used is not bounded[7], thus reaching a local maximum does not guarantee that a higher utility cannot be obtained. Therefore, an algorithmic approach that can reduce the complexity of extracting the optimal cooperative parallel search is favorable.

5. Algorithmic Approach

In this section we present a comprehensive analysis of the problem, emphasizing some unique characteristics of the coalition's optimal strategy. These findings lead to an algorithm for computing $V(s,w)$ with a polynomial complexity in the number of potential interactions, w (which is the key component for computing $x^*(s)$ and $V^*(s)$).

7. The number of opportunities is theoretically infinite due to the high arrival rate of new opportunities as derived from such a dynamic environment.

5.1 Reducing Calculation Complexity

Recall that when attempting to solve the problem as a set of equations (see section 4) the potential number of parallel interactions that may be used is unbounded. Nevertheless, in order to extract $x^*(s)$ it is essential to supply the coalition with an upper bound, w^s_{max}, for the optimal number of parallel interactions to be used when in state s. In order to overcome this difficulty and to suggest an upper bound for w^s_{max} we make use of the following notation. We use $S_s = (s_1, ...s_{|S_A|})$ to denote the states constituting S_A, sorted by their termination utilities $V_t(s)$[8], where s_1 is the state with the highest expected utility $V_t(s)$ in S_A.
The following proposition suggests an efficient upper bound for $x^*(s)$.

Proposition 1. *For each state s_i an upper bound $w^{s_i}_{max}$, to $x^*(s_i)$ can be calculated using $w^{s_i}_{max} = \lceil w \rceil$, where w is the solution to the following equation:*

$$c(w,n) = V_t(s_1) - V_t(s_i) \tag{7}$$

The suggested bound is valid simply because for every value of w greater than $w^{s_i}_{max}$ the search cost associated with the following immediate search round $c(w,n)$ is greater than any possible future improvement in the coalition's utility $V_t(s_1) - V_t(s_i)$ (since the maximum additional utility that the coalition can gain is bounded by the difference between the coalition's overall maximum utility $V_t(s_1)$ and its immediate utility from its current state $V_t(s_i)$). Later on, we show that the above upper bound value is a byproduct of the main loop in our proposed algorithm, thus it does not even need to be directly calculated.

Having an upper bound for $x^*(s)$ is an important step towards a solution, however the calculation of $V(s,w)$ (from which $x^*(s)$ can be derived) is still exponential in the number of parallel interactions used, w. Our analysis, which is based on the restructuring of the different elements composing $V(s,w)$, allows us to bypass this complexity through the introduction of a finite algorithm with a polynomial computational complexity in w that will inevitably identify the optimal strategy for the coalition.

In order to reduce the complexity of computing the coalition's best strategy, we make use of the following notations and definitions:

- We use $p^{stay}(s,w)$ to denote the probability the coalition will remain in the same state s after conducting w parallel interactions. This can be calculated as the probability of none of the encountered w opportunities changing the coalition's state:

$$p^{stay}(s,w) = (p^{stay}(s,1))^w = \Big(\sum_{\{\vec{o}\} \in \Theta^s_{stay_1}} p(\vec{o})\Big)^w \tag{8}$$

  The term $1-(p^{stay}(s,1))^w$ can now be used as a better structured representation of the element $\sum_{\theta_w \in \Theta^s_{improve_w}} p_w(\theta_w)$ in Equation 6.

- We use $V^{new}(s,k)$ to denote the coalition's expected utility obtained from potentially reaching new states (e.g. different than s) after maintaining k parallel interactions, while using the optimal strategy $x^*(s')$ for each new future state s'. The term $V^{new}(s,k)$ does not take into account the cost associated with the current k interactions. However it does consider the

8. If there are several states with equal utility they are sorted according to a pre-defined order.

search costs associated with any further search stages, originating in new states. Notice that $V^{new}(s,k)$ is equal to zero if the coalition remains in state s after k interactions. The term $V^{new}(s,k)$ can be expressed as:

$$V^{new}(s,k) = \left\{ \begin{array}{ll} \sum_{\theta_k \in \Theta^s_{improve_k}} p_k(\theta_k) V^*(state(s \cup \theta_k)) & k > 0 \\ 0 & \text{k=0} \end{array} \right\}. \tag{9}$$

Note that $V^{new}(s,w)$ is actually one of the elements in Equation 6. Therefore, Equation 6 can now be formulated as ($\forall w > 0$):

$$V(s,w) = \frac{-c(w,n) + V^{new}(s,w)}{1 - p^{stay}(s,1)^w} \tag{10}$$

The calculation of $V^{new}(s,w)$ using Equation 9 is still exponential in the number of the parallel interactions. In order to efficiently compute $V^{new}(s,w)$ in Equation 10 we consider the w simultaneous interactions as w sequential interactions, associated with no search costs. This fully complies with the definition of $V^{new}(s,w)$ as given above (as the cost of the w interactions is already considered). The justification for the above representation method is given in the Lemma 1 which follows directly from Theorem 1.

Lemma 1. *A new state reached by obtaining a new set of opportunities is equivalent to a state reached by sequentially obtaining pairwise disjoint subsets of this set. Formally stated, given a set* θ_w *and any number of subsets* $\theta^1_{w_1}, \dots, \theta^r_{w_r}$, $\theta^i_{w_i} \subseteq \theta_w$, $\theta^i_{w_i} \cap \theta^j_{w_j} = \emptyset, i \neq j$, $\theta^1_{w_1} \cup \dots \cup \theta^r_{w_r} = \theta_w$ *and a state s, then the following holds:*

$$state(s, \theta_w) \equiv state(state(\dots state(state(s \cup \theta^1_{w_1}) \cup \theta^2_{w_2}) \dots \cup \theta^{r-1}_{w_{r-1}}) \cup \theta^r_{w_r}) \tag{11}$$

Proof. We begin by proving the following supporting lemma:

Lemma 2. *Let* $\theta', \theta'' \in \Theta$ *be two sets of opportunities. If* $\theta' \equiv \theta''$ *then* $\forall \theta \in \Theta$, $state(\theta' \cup \theta) \equiv state(\theta'' \cup \theta)$.

Proof. Since $\theta' \equiv \theta''$, then from the definition of the equivalence relation it follows that $\theta' \cup \theta \equiv \theta'' \cup \theta$. From Theorem 1 and the *state* definition it follows that $\theta' \cup \theta \equiv \{alloc(\theta' \cup \theta)\} = state(\theta' \cup \theta)$ and $\theta'' \cup \theta \equiv \{alloc(\theta'' \cup \theta)\} = state(\theta'' \cup \theta)$. From the transitive characteristic of the equivalence relation it follows that $state(\theta' \cup \theta) \equiv state(\theta'' \cup \theta)$. ☐

Let s be a state and let θ_w be a set of opportunities. We prove Lemma 1 by induction on the number of disjoint sets r. For $r = 2$, let $\theta'_{w'}$ and $\theta''_{w''}$ be any sets that satisfy $\theta'_{w'} \cup \theta''_{w''} = \theta_w$ and $\theta'_{w'} \cap \theta''_{w''} = \emptyset$. From Theorem 1 and the *state* definition we obtain:

$$s \cup \theta'_{w'} \equiv \{alloc(s \cup \theta'_{w'})\} = state(s \cup \theta'_{w'}) \tag{12}$$

From 12 and Lemma 2 it follows that

$$state(s \cup \theta_w) = state(s \cup \theta'_{w'} \cup \theta''_{w''}) \equiv state(state(s, \theta'_{w'}) \cup \theta''_{w''}) \tag{13}$$

We assume that for any number of disjoint sets $r \leq M$ and $r > 2$, $\theta^1_{w_1}, \dots, \theta^r_{w_r}$, $\theta^i_{w_i} \subseteq \theta_w$, $\theta^i_{w_i} \cap \theta^j_{w_j} = \emptyset, i \neq j$, $\theta^1_{w_1} \cup \dots \cup \theta^r_{w_r} = \theta_w$ Equation 11 holds and we prove that Equation 11 holds for r=M+1.

Given θ_w and sets $\theta^1_{w_1},...,\theta^r_{w_r}$, we can decompose θ_w to two disjoint sets θ_{w-w_r} and $\theta^r_{w_r}$, where $\theta_{w-w_r} = \theta_w \setminus \theta^r_{w_r}$. Therefore as we proved for $r = 2$ the following holds:

$$state(s \cup \theta_w) \equiv state(state(s, \theta_{w-w_r}) \cup \theta^r_{w_r}). \tag{14}$$

In addition we can decompose the set θ_{w-w_r} to $r-1 \leq M$ sets $\theta^1_{w_1},...,\theta^{r-1}_{w_{r-1}}$. Therefore using the induction assumption:

$$state(s \cup \theta_{w-w_r}) \equiv state(...state(state(s, \theta^1_{w_1}) \cup \theta^2_{w_2})... \cup \theta^{r-1}_{w_{r-1}}) \tag{15}$$

From Lemma 2 and 15 we obtain:

$$state(state(s \cup \theta_{w-w_r}) \cup \theta^r_{w_r}) \equiv state(state(...state(state(s \cup \theta^1_{w_1}) \cup \theta^2_{w_2})... \cup \theta^{r-1}_{w_{r-1}}) \cup \theta^r_{w_r}) \tag{16}$$

From 14, 16 and the transitive characteristic of the equivalence relation it follows that:
$state(s \cup \theta_w) \equiv state(state(...state(state(s \cup \theta^1_{w_1}) \cup \theta^2_{w_2})... \cup \theta^{r-1}_{w_{r-1}}) \cup \theta^r_{w_r})$ □

A specific case of the above Lemma 1 is when each subset consists of a single opportunity. Thus the calculation of $V^{new}(s,k)$ can recursively rely on the values $V^{new}(s',k-1)$, where s' represents any of the new states reached after obtaining one additional opportunity from O_p. Therefore in order to compute $V^{new}(s,k)$, we merely consider the expected utility after conducting one interaction from the planned k interactions. Here, with a probability of $p^{stay}(s,1)$ the coalition remains in the same state s, where the expected utility (having $k-1$ additional interactions to go) is $V^{new}(s,k-1)$ (recall that $V^{new}(s,0) = 0$ according to Equation 9). Otherwise, if a new state s' is reached after a single interaction, then there is the possibility of reaching further new states in the remaining $k-1$ interactions (taking these states' utility into consideration by the term $V^{new}(s',k-1)$) or with a probability of $p^{stay}(s',k-1)$ remaining in the new state s' even after the additional $k-1$ interactions (in which case the utility is the one obtained by resuming the search from this state and on using the optimal strategy, $V^*(s')$). The above description is encapsulated in the following recursive equation:

$$V^{new}(s,k) = \left\{ \begin{array}{ll} p^{stay}(s,1)V^{new}(s,k-1) + \sum_{\{\vec{o}\} \in \Theta^s_{improve_1}} p(\vec{o})(V^{new}(s',k-1) + p^{stay}(s',k-1)V^*(s')) & k > 0 \\ 0 & \text{k=0} \end{array} \right\} \tag{17}$$

where $s' = state(s \cup \vec{o})$. Notice that when repeating the calculation using the above equation with an increasing k value starting from $k = 1$, each iteration includes only a single unknown parameter, ($V^{new}(s,k)$). In addition, the values $V^{new}(s',k-1)$ and $V^*(s')$ do not depend on values that were computed for state s (s' precedes s in the set S_s).

5.2 The Algorithm for Computing the Coalition's Optimal Strategy

The above analysis leads to an algorithm for computing the coalition's optimal strategy (Algorithm 1). This algorithm computes the coalition's strategy and its expected utility for all possible states S_A. Its execution time is polynomial in w^{init}_{max} (for convenience we use w^{init}_{max} to denote the bound for the number of parallel interactions w, when the coalition begins its search, i.e., coalition's state is $s = s_{init} = \{\}$).

Notice that at each stage of its execution, algorithm 1 reuses components computed in earlier stages. For example, $V^{new}(s,w)$ appears both in the computation of $V(s,w)$ (using Equation 10), in the computation of $V^{new}(s',w+1)$ (using Equation 17, when $\exists \vec{o}' \in O_p$ such that $state(s' \cup \vec{o}') = s$)

Algorithm 1 An algorithm for computing the optimal search strategy x^*

Input: O_p - set of potential opportunity types in the market; $p(\vec{o})$ - opportunity types' probability function; n - coalition's size; $U_j(),\ j = 1,...,n$ - coalition members' utility functions; $c(w,n)$ - search cost function; S_A - set of all possible states of a coalition A.

Output: $x^*(s)\ \forall s \in S_A$ - the coalition's optimal strategy.

1: Build the set of ordered states S_s
2: **for** i=1 to $|S_s|$ **do**
3: Set $V(s_i,0) = V_t(s_i)$ using Equation 1
4: Set $V^{new}(s_i,0) = 0$
5: Set $w = 1$
6: **while** $c(w,n) \leq (V_t(s_1) - V_t(s_{init}))$ **do**
7: Compute $V^{new}(s_i,w)$ using Equation 17
8: Compute $V(s_i,w)$ using Equation 10
9: w++;
10: **end while**
11: Set $x^*(s_i) = \arg\max_{w' \in (0,...,w-1)} V(s_i,w')$, $V^*(s_i) = V(s_i,x^*(s_i))$
12: **end for**
13: Return $(x^*(s_i),\ i = 1,...,s_{|S_A|})$

and in the computation of $V^{new}(s'',w+1)$ (using Equation 17, when $\exists \vec{o}'' \in O_p$ such that $state(s'' \cup \vec{o}'') = s$). Storing the result for each such computational element in the memory, for the purpose of reusing it at later stages, significantly improves the efficiency of the algorithm. This is accomplished by using two matrixes V and V^{new} of size $|S_A| \times (w^{init}_{max}+1)$, in which the corresponding $V(s,w)$ and $V^{new}(s,w)$ values are stored for each pair (s,w), representing a state and the correlated result of each number of parallel interactions that is used for the calculations. Additionally, we store x^* and V^* values by using two arrays of size $|S_A|$ for reusing $x^*(s')$ and $V^*(s)$ in the computation of $V^{new}(s,w)$.

Theorem 2. *Algorithm 1 returns the optimal strategy of the coalition in a polynomial time of w^{init}_{max}.*

Proof. In our proof, we assume that we use the matrices V and V^{new} for storing the values computed along the execution of the algorithm as described above. In order to build the set of ordered states S_s, the algorithm needs to compute the coalition's termination utility for each of the states in S_A, according to equation 1. Computing the coalition's termination utility for a given state takes $O(|A|^3)$ in C2C markets[9] and $O(|A|^2)$ in B2C markets[10]. The coalition's termination utility is calculated for each state $s \in S_A$. Thus the overall complexity of computing the coalition's termination utility for all states is $O(|S_A||A|^2)$ and $O(|S_A||A|^3)$ in B2C and C2C markets, respectively (the sets S_s and S_A have the same size). Sorting the members in S_A in order to build S_s takes $O(|S_A|\log|S_A|)$. The computation of step 3 takes a constant time assuming that we stored for each

9. As mentioned previously, computing the coalition's termination utility in C2C markets is equivalent to finding a maximum matching in a weighted bipartite graph. Finding maximum matching in a weighted bipartite graph can be done in $O(n(n\log n+m))$, where n is the number of vertices and m is the number of edges (Wang, Makedon, & Ford, 2004). Since in our graph each agent is connected to all opportunities, and the number of opportunities in a given state is bounded by the number of agents, the process of computing the coalition's termination utility takes $O(|A|^3)$.

10. As mentioned previously, computing the coalition's termination utility in B2C markets can be done by finding an opportunity for each agent in the coalition that maximizes its utility. As the number of opportunities in a given state is bounded by the number of agents, computing the coalition's termination utility takes $O(|A|^2)$.

state its coalition's termination utility when S_s was built. Steps 3-5 are performed for all states, thus, the overall complexity is $O(|S_A|)$. The loop in step 6 is performed at most w_{max}^{init} times. This is because when reaching a value $w = w_{max}^{init} + 1$ in step 6, the loop condition no longer holds (i.e., $c(w,n) > (V_t(s_1) - V_t(s_{init}))$). Notice that the first elements calculated are for state s_1 (i.e. the one with the maximum utility) according to the loop in step 2. Here, as explained in section 4, the expected utility from any strategy by which the search is resumed (i.e., using $w \geq 1$) is $V(s,w) = -\infty$ (formally, since $\Theta_{improve_w}^{s_1} = \emptyset\ \forall w$ then $p^{stay}(s_1,w) = 1$ and $V^{new}(s_1,w) = 0\ \forall w \in (1,..,w_{max}^{init})$, thus, $V(s_1,w) = -\infty\ \forall w \in (1,..,w_{max}^{init})$). Therefore, the optimal strategy in state s_1 is to terminate the search, i.e. $x^*(s_1) = 0$ and $V(s_1,0) = V_t(s_1)$. For any other state, s, when reaching step 7, the algorithm has already computed both $V^*(s')$ and $V^{new}(s',w)\ \forall w \in (0,..w_{max}^{init})$ for each potential future new state s' originating from state s. This is due to the fact that all future states of state s appear before s in the set of ordered states S_s (either because of a higher utility, or equal utility yet sorted before s according to the function $alloc$). In addition, for any number of parallel interactions $w \geq 1$, the algorithm has already computed $V^{new}(s,w-1)$. Therefore, the computation time in step 7 sums up to an order of $|O_p|$ components that have already been computed. Then, when reaching step 8, the value of V^{new} (which is part of the numerator in equation 10) has been already computed. Therefore, if we ignore, for now, the time of computing the coalition's new state s' when it encounters opportunity $\vec{o}$ given it is in a state s and the time of computing p^{stay} values, the computation of step 7 takes $O(|O_p|)$ and the computation of step 8 takes a constant time. Since these steps are performed at most $w_{max}^{init}|S_A|$ times, the overall complexity of computing these steps is $O(|O_p|w_{max}^{init}|S_A|)$. In step 11, the algorithm chooses the maximum value among w_{max}^{init} values that have already been computed. Since this step is performed for each state $s \in S_s$, the overall complexity of performing this step is $O(w_{max}^{init}|S_A|)$. The complexity of computing the coalition's new state s' when it encounters opportunity $\vec{o}$ given that its current state is s depends on the market type. In B2C markets, the complexity is $O(|A|)$ if we store for each state s also its $alloc(s)$ value (we compute this value when we compute the coalition's termination utility). In C2C markets, computing the new state takes $O(|A|^3)$[11]. We can store these values using a matrix S_{future} of size $|S_A| \times |O_p|$ in which the new state $s' = state(s \cup \vec{o})$ is stored for each pair $(s,\vec{o})$. Therefore, the overall complexity of computing future states for all states is $O(|S_A||A||O_p|)$ in B2C markets, and $O(|S_A||A|^3|O_p|)$ in C2C markets. Computing $p^{stay}(s,w)$ can be done in a constant time based on $p^{stay}(s,w-1)$. Moreover, we can store these values using a matrix P_{stay} of size $|S_{A_g}| \times w_{max}^{init}$ in which the value $p^{stay}(s,w)$ is stored for each pair (s,w). Therefore, the total complexity of computing p^{stay} values is $O(|S_A| \times w_{max}^{init})$. Given this analysis, the overall complexity of the algorithm is $O(|S_A||A|^2 + |S_A|\log|S_A| + |S_A||O_p|w_{max}^{init} + |S_A||A||O_p|)$ in B2C markets, and $O(|S_A|\log|S_A| + |S_A||A|^3|O_p| + |S_A||O_p|w_{max}^{init})$ in C2C markets. Hence the algorithm is polynomial in w_{max}^{init}. $\square$

Note the algorithm uses w_{max}^{init} as an upper bound for the optimal number of interactions $x^*(s_i)$, $\forall i = 1,...,|S_s|$. This bound is valid since according to Equation 7 the value of $w_{max}^{s_i}$ increases as $V_t(s_i)$ decreases and the coalition's termination utility reaches its minimum in the initial state ($V_t(s_{init}) = V_t(\{\}) = 0$). This suggests a further significant improvement of the above algorithm. Instead of using w_{max}^{init} for all states, we calculate a specific upper bound, $w_{max}^{s_i}$, for each state s_i in step 6, according to Equation 7. At some point this may require re-computation of $V(s_i,w)$ for w

11. Computing the coalition's new state in C2C markets is done by finding a maximum matching in a weighted bipartite graph (see footnote 10 for complexity analysis).

values that were not used in previous execution stages of the algorithm; however, the total number of calculations for each state s_i in many cases will significantly decrease[12].

6. Properties of the Parallel Cooperative Search (PCS) Model

Prior to extending the above analysis to scenarios where the coalition faces a finite decision horizon, we emphasize some interested unique characteristics of the parallel cooperative search derived from the general analysis. First we wish to emphasize that the PCS model is a generalization of both the Single Agent Parallel Search (SAPS) and the Sequential Cooperative Search (SCS) model as stated in the following proposition.

Proposition 2. *The cooperative parallel search is a generalization of both the single agent parallel search and the cooperative sequential search models*[13].

Proof. From the analysis given in section 4 it is clear that Algorithm 1 results in the same strategy used in the cooperative sequential search and in the single agent's parallel search, for the specific cases in which $\forall s_i \; w_{max}^{s_i} = 1$ parallel interactions or $n = 1$ agents, respectively.[14] $\square$

Furthermore, we emphasize that the coalition's expected utility never decreases when using our proposed mechanism in comparison to the sequential cooperative search. Indeed in the case where maintaining more than a single interaction is not a favorable strategy Algorithm 1 results in one interaction at a time strategy, as used in the sequential cooperative search. Obviously if using the parallel interaction does not decrease the search cost (i.e. the search cost of conducting any number of interactions sequentially is equal or smaller than conducting the interactions in parallel), then the sequential cooperative search is the dominating strategy as stated in the next proposition.

Proposition 3. *If the cost of conducting any number of parallel interactions is equal or higher than the cost of conducting these interactions sequentially (i.e $\forall w,\; w * c(1,n) \leq c(w,n)$) then the use of the parallel search can at most match the expected utility of the sequential cooperative search.*

Proof. Consider an optimal cooperative parallel search strategy x^*_{par}. Obviously there is a state $s \in S_A$ satisfying: (1) The coalition's strategy in s is to conduct more than one interaction, $x^*_{par}(s) > 1$ (2) For any future state s' of s the coalition conducts at most one interaction (or terminates search), i.e., $x^*_{par}(s') \leq 1$. If we replace the parallel search strategy when in state s with the sequential search from this state on then the expected utility can only increase. This is because in the worst case scenario the coalition will execute w interactions incurring at most the same cost as in x^*_{par}, ending up with the same expected termination utility. For any other case, whenever the coalition reaches state s' where its strategy according to x^*_{par} is to terminate its search prior to completing w interactions, the expected utility is at least the expected utility achieved by using x^*_{par} (otherwise the coalition's strategy in s' according to x^*_{par} will be to resume the search) . Therefore the use of the new strategy can only improve the expected utility. Using backward induction we can apply the same logic to all former states for which x^*_{par} implies more than one interaction in parallel.

12. The extent of the achieved improvement is highly correlated with the specific environment in which the coalition operates.
13. Notice that in this context the single agent sequential search is a specific case of the single agent parallel search, where the agent interacts with a single seller agent at a time.
14. Assuming similar cost structures in all three models.

Consequently, we obtain that the sequential strategy results in at least as large an expected utility as x^*_{par}. □

Next we consider the case where the agents are fully homogeneous (in terms of their utility functions) and operate in B2C markets. Here, we can prove that the optimal strategy of the coalition is stationary (i.e. does not change according to the current state). Furthermore, we show that the stationary strategy characteristic holds not only for fully homogeneous agents but also when the agents have correlated preferences. Two agents a_i and a_j have correlated preferences when agent a_i prefers $\vec{o}'$ over $\vec{o}''$ if and only if agent a_j prefers opportunity $\vec{o}'$ over $\vec{o}''$ (i.e $\forall \vec{o}', \vec{o}'' \in O_p$, $U_j(\vec{o}') \leq U_j(\vec{o}'') \leftrightarrow U_i(\vec{o}') \leq U_i(\vec{o}'')$) and vice versa.

Theorem 3. *In the B2C market, if all the agents have correlated preferences, the search strategy is based on a reservation value*[15] U_{rv}*. In such a scenario the number of parallel interactions the coalition uses according to the optimal strategy (in case it resumes the search) is fixed during the search (*$\forall s \in S_A$ *such that* $V_t(s) < U_{rv}$ *exists* $x^*(s) = w_{fixed}$*).*

Proof. In the above scenario, the search problem is equivalent to the problem of a single agent with a utility function equal to the sum of the different agents' utilities, $U = U_1 + U_2 + \ldots + U_n$. In this case after terminating the search, all of the coalition members are always assigned the same opportunity. Therefore, the search strategy is reservation value based, and the search is terminated upon reaching such an opportunity with a utility exceeding a pre-set reservation value U_{rv}. Since the probability of reaching such an opportunity does not depend on the coalition's state, the number of parallel interactions used throughout the search is fixed. □

Nevertheless Theorem 3 does not hold in the C2C market even when the agents are homogeneous as the next lemma states.

Lemma 3. *In the C2C market even for fully homogeneous agents (all agents have the same utility functions) the search strategy is not always stationary and the optimal number of parallel interactions can be changed during the search according to the coalition's current state.*

Proof. In order to prove this lemma, we consider a coalition that operates in a C2C market in the following environment:

Environment 4. *Assume a coalition of two agents,* a_1 *and* a_2*, searching for a product (e.g., a used book) characterized by one attribute (e.g., indicating whether the book is signed by the author) with two possible values, 1 (signed) and 2 (not signed). This results in two opportunity types,* $\vec{o}_1 = (1)$ *and* $\vec{o}_2 = (2)$*. Assume opportunity* $\vec{o}_1$ *is rare and can be found with a probability of* $1/100$*, while opportunity* $\vec{o}_2$ *is very common (a probability of* $99/100$*). Both agents' utilities are* 100 *for the rare opportunity and* 1 *for the common opportunity. Consider that the cost of conducting* w *parallel interactions is equal to the cost of conducting* w *interactions sequentially,* $c(w,n) = wc(1,n)$*, where the cost of conducting a single interaction is* $c(1,n) = 0.1 + 0.05n$*,* $\forall (n > 1)$*. Table 4 summarizes environment's setting.*

We consider the two following states: (1) The coalition starts its search (the coalition's current state is its initial state $s = \{\}$) (2) The coalition encounters opportunity $\vec{o}_1$ (the coalition's current

15. A reservation value is a value that the coalition sets a-priori and terminates the search if reached an opportunity associated with a utility greater than or equal to this value (Sarne & Kraus, 2005).

Opportunity	*Attribute*	*Probability*	*Utility*	
			Agent a_1	*Agent a_2*
$\vec{o}_1$	signed	0.01	100	100
$\vec{o}_2$	not signed	0.99	1	1

Table 4: Agents' utilities for the four opportunities in Environment 4

state is $s' = \{\vec{o}_1\}$).
In order for the coalition to terminate its search all its members should exploit opportunity $\vec{o}_1$ (otherwise the expected utility from resuming the search exceeds the search cost). Thus in order for the coalition to terminate the search in the former case (the coalition's state is $s = \{\}$) it must encounter opportunity $\vec{o}_1$ twice and in the latter case (the coalition's state is $s' = \{\vec{o}_1\}$) it has to encounter opportunity $\vec{o}_1$ once. Thus we expect that the number of optimal interactions, where the coalition's state is s, to be different from the number of optimal interactions, where its state is s'. Indeed the computation of the coalition's optimal number of interactions (using Algorithm 1) results in $x^*(s) = 299$ and $x^*(s') = 161$. □

7. Finite Decision Horizon

An important variant of the cooperative parallel search is the one where the agents forming the coalition are restricted by a deadline for finalizing their search. For example, consider a coalition that searches for costumes for its members to wear to a costume party. In this case the coalition can not search for the costumes forever since the customs have a value for the coalition members only if purchased prior to the party. This type of environment is often referred to in search theory as a "finite decision horizon environment". Specifically, within the context of this paper we consider finite decision horizon environments where the coalition as a whole should terminate its search prior to or within the next r search rounds. Note that in the sequential search this definition is equivalent to the definition stating that the coalition can conduct at most r additional interactions.

In addition to the general advantages recognized in the parallel cooperative search, in finite decision horizon environments the model can enable the coalition to conduct more than the maximum r interactions facilitated by the sequential model (whenever necessary). Moreover, in some environments where the coalition cannot improve its performance by interacting with several sellers in parallel (e.g., in cases such as those described in Proposition 3), the introduction of a finite decision horizon constraint creates a strong incentive to interact with more than a single seller at a time. This is illustrated in the following example:

Example 3. *Consider a coalition that operates in a B2C market, having the same characteristics as in Environment 4, which was introduced in section 6. In this environment the coalition's optimal strategy when having an infinite decision horizon is to search sequentially (see Proposition 3). Nevertheless, for various finite decision horizon values the coalition can benefit from using the parallel search technique. The most trivial example in this case is when the coalition has to terminate its search within the next search round. In this case, if the coalition conducts several interactions in the next round, its probability to encounter opportunity $\vec{o}_1$ increases in comparison to the case in which it conducts only a single search. For example if the coalition conducts 100 interactions it has a probability of* 0.99^{100} *to encounter only opportunity $\vec{o}_2$ during its search and a probability of*

$1-0.99^{100}$ *to encounter opportunity* $\vec{o}_1$ *during its search and its expected utility can be expressed as:* $(1-0.99^{100})*V_t(\{\vec{o}_1\})+0.99^{100}V_t(\{\vec{o}_2\})-c(100,2)=(1-0.99^{100})*200+(0.99)^{100}*2-0.2*100=107.52$. *This value is significantly greater than its expected utility when using the sequential cooperative search which is* $0.01*V_t(\{\vec{o}_1\})+0.99V_t(\{\vec{o}_2\})-c(1,2)=3.78$.

In order to compute the coalition's strategy in a finite decision horizon model variant we extend our definitions to include the number of remaining search rounds, r. We use $V(s,w,r)$ to denote the coalition's expected utility, when it conducts w interactions in the next round and has to terminate its search within the next r rounds. The term $V(s,0,r)$ denotes the immediate utility obtained, if the coalition decides to terminate the search at state s, thus: $V(s,0,r)=V_t(s)$. We denote by $x^*(s,r)$ and $V^*(s,r)$ the coalition's optimal strategy and its expected utility (when using its optimal strategy) at state s, if it should terminate its search within the next r rounds.

The $V^*(s,r)$ and $x^*(s,r)$ calculation is similar to the infinite decision horizon case, except that whenever the coalition reaches its decision deadline it inevitably terminates its search. Namely, if $r=0$, then $x^*(s,0)=0$ and the coalition's expected utility $V^*(s,0)$ is equal to $V_t(s)$.

$$x^*(s,r)=\begin{cases}\arg\max_w V(s,w,r) & r>0\\ 0 & \text{r=0}\end{cases} \tag{18}$$

$$V^*(s,r)=V(s,x^*(s),r) \tag{19}$$

We begin by computing $V(s,w,r)$, when $r>0$ (since $V(s,w,0)=0$, $\forall w$). Here, we can apply the same analysis methodology used in section 4. However, the expected utility of the coalition when resuming its search following the current search stage should reflect the change in the decision horizon. Therefore, instead of using Equation 4, we should use the following modification:

$$V(s,w,r)=-c(w,n)+\sum_{\theta_w\in\Theta^s_{improve_w}} p_w(\theta_w)V^*(s',r-1)+p^{stay}(s,w)V^*(s,r-1),\quad \forall w>0,r>0 \tag{20}$$

where $s'=state(s\cup\theta_w)$.

The computation of Equation 20 is exponential in the number of parallel interactions. In order to reduce the complexity of V(s,w,r) for $r>0$ we consider the two following cases:

- $w=1$: the coalition encounters a single opportunity thus Equation 20 can be expressed as:

$$V(s,1,r)=-c(1,n)+\sum_{\{\vec{o}\}\in\Theta^s_{improve_1}} p(\vec{o})V^*(state(s\cup\vec{o}),r-1)+\sum_{\{\vec{o}\}\in\Theta^s_{stay_1}} p(\vec{o})V^*(s,r-1),\quad \forall r>0 \tag{21}$$

  In this case, no additional computational complexity is introduced, in comparison to the SCS model (Sarne & Kraus, 2005).

- $w>1$: here we attempt to find a computational means for extracting the value of the term $\sum_{\theta_w\in\Theta^s_{improve_w}} V^*(s',r-1)+p^{stay}(s,w)V^*(s,r-1)$ (in Equation 20) in complexity polynomial in w. This expression denotes the coalition's expected utility when it conducts w interactions without considering the cost associated with conducting the w interactions. In order to efficiently compute this value we consider the coalition's expected utility after conducting one of its w interactions and obtaining opportunity $\vec{o}$. In this case, the coalition's expected utility is equal to the coalition's expected utility when starting from state $state(s \cup \vec{o})$ and conducting $w-1$ interactions plus a cost equivalent to the cost of maintaining these $w-1$ interactions, $c(w-1,n)$ (these are added since they were already subtracted from the expected utility in Equation 20 and $V(state(s \cup \vec{o}), w-1, r)$ subtracts them again):

$$V(s,w,r) = -c(w,n) + \sum_{\vec{o} \in O_p} p(\vec{o})(V(state(s \cup \vec{o}), w-1, r) + c(w-1,n)), \quad \forall w > 1, r > 0 \tag{22}$$

The sum in Equation 22 can be represented as the sum of the expected utility over the opportunities that change the coalition's current state and the sum over the opportunities that do not change the coalition's current state:

$$V(s,w,r) = -c(w,n) + \sum_{\{\vec{o}\} \in \Theta^s_{improve_1}} p(\vec{o})V(s', w-1, r) + \sum_{\{\vec{o}\} \in \Theta^s_{stay_1}} p(\vec{o})V(s, w-1, r) + c(w-1,n), \quad \forall w > 1, r > 0 \tag{23}$$

where $s' = state(s \cup \vec{o})$.

Equations 23 and 21 facilitate the calculation of $V(s,w,r)$ in a polynomial time of w. In order to find the coalition's optimal strategy, the efficient bound for the optimal number of interactions given in Equation 7 is also valid for the finite decision horizon variant.

The above analysis leads to Algorithm 2 which is a modification of Algorithm 1. This algorithm computes the coalition's strategy and expected utility for all states in S_A. In order to compute $V^*(s,r)$, this algorithm uses backward induction. It starts by computing $V^*(s_i, 0)$ for all states. Here, the coalition is forced to terminate its search thus the algorithm sets the expected utility to be equal to the termination utility: $V^*(s_i, 0) = V_t(s_i)$. Then, the algorithm computes $V(s_i, r, w)$ for $r > 0$ using Equation 21 and 23 starting from $w = 1$ till $c(w,n) \geq V_t(s_1) - V_t(s_{init})$. The algorithm sets the coalition's optimal strategy to be the number of interactions w that maximizes $V(s_i, r, w)$.

Similar to the infinite decision horizon case, Algorithm 2 reuses components computed in earlier stages in each stage of its execution. Nevertheless, in this case in order to store the results in the memory for reuse later, we need a three dimensional matrix V of size $|S_A| \times (w^{init}_{max} + 1) \times (r_{max} + 1)$ where r_{max} is the initial decision horizon. The corresponding $V(s,w,r)$ values are stored for each triplet (s,w,r), representing a coalition's state, the number of simultaneous interactions used and the limit on the number of rounds that the coalition can conduct. Additionally, we store $V^*(s,r)$ and $x^*(s,r)$ values in two matrixes V and X of size $|S_A| \times (r_{max} + 1)$ for reusing $x^*(s', r-1)$ and $V^*(s', r-1)$ in the computation of $V(s,w,r)$.

In the following theorem we prove that algorithm 2 returns the coalition's optimal strategy and its expected utility, in a polynomial time of w^{init}_{max} (the bound on the number of parallel interactions computed using Equation 7) and r_{max}.

Theorem 4. *Algorithm 2 returns the optimal strategy of the coalition and its expected utility from following this strategy, where it should terminate its search within the next r_{max} rounds, in a polynomial time of w^{init}_{max} and r_{max}.*

Proof. In our proof, we assume the use of matrices for storing the computed values as described above. The process of extracting S_s can be done exactly as in the finite decision horizon. Thus, step 1 is of complexity $O(|S_A||A|^2 + |S_A|\log|S_A|)$ and $O(|S_A||A|^3 + |S_A|\log|S_A|)$ for B2C markets, and in

Algorithm 2 An algorithm for computing the optimal search strategy x^* when the coalition should terminate its search within the next r_{max} rounds

Input: O_p - set of potential opportunity types in the market; $p(\vec{o})$ - opportunity types' probability function; n - coalition's size; $U_j(),\ j=1,...,n$ - coalition members' utility functions; r_{max} - the maximum number of search rounds until terminating the search (the decision horizon); $c(w,n)$ - search cost function; S_A - set of all possible states of a coalition A.

Output: $x^*(s,r), V^*(s,r)\ \forall s \in S_s,\ \forall 0 \leq r \leq r_{max}$ - the coalition's optimal strategy and its expected utility.

1: Build the set of ordered states S_s
2: **for** $i = 1$ to $|S_s|$ **do**
3:    Set $x^*(s_i, 0) = 0$
4:    Set $V^*(s_i, 0) = V_t(s_i)$ using Equation 1
5: **end for**
6: **for** $r = 1$ to r_{max} **do**
7:    **for** i=1 to $|S_A|$ **do**
8:       Set $V(s_i, 0, r) = V_t(s_i)$ using Equation 1
9:       Compute $V(s_i, 1, r)$ using Equation 21
10:       Set $w = 2$
11:       **while** $c(w,n) \leq (V_t(s_1) - V_t(s_{init}))$ **do**
12:          Compute $V(s_i, w, r)$ using Equation 23
13:          w++;
14:       **end while**
15:       Set $x^*(s_i, r) = \arg\max_{w' \in (0,...,w-1)} V(s_i, w', r)$, $V^*(s_i, r) = V(s_i, x^*(s_i, r), r)$
16:    **end for**
17: **end for**
18: Return ($x^*(s_i, r)$, $i = 1,...,s_{|S_A|}$, $r = 0,...,r_{max}$)

C2C markets, respectively. The computation in steps 3 and 4 takes a constant time, assuming that we stored for each state its coalition's termination utility when we built S_s. These steps are performed for all states, thus, the overall complexity of computing these steps throughout the loop is $O(|S_A|)$. Again, step 8 can be performed in a constant time, assuming the coalition's termination utility is stored for each state. For any state, s, and for any limit on the number of search rounds $r > 0$ when reaching step 9 of the algorithm, the algorithm has already computed $V^*(s', r-1)$ for each potential future new state s' originating from state s. Therefore, the computation in step 9 is simply the summing of order of $|O_p|$ components that have already been computed. Consequently, if we ignore the time of computing the coalition's new states s' (we will add the time for computing these values later), the computation of this step takes $O(|O_p|)$. Since steps 8-9 are performed $r_{max}|S_A|$ times, the overall complexity of computing these steps (when ignoring the time of computing the new states) is $O(|O_p| r_{max} |S_A|)$. Notice that when $w = w^{init}_{max} + 1$ we obtain $c(w,n) > (V_t(s_1) - V_t(s_{init}))$. Thus, the loop in step 11 is performed at most w^{init}_{max} times. For any $w > 0$ when reaching step 12, the algorithm has already computed $V(s', w-1, r)$ for each potential future new state s' originating from state s. Therefore, the computation in step 12 takes $O(|O_p|)$ when ignoring the time of computing the new states. Since this step is performed $r_{max}|S_A|w^{init}_{max}$ times, the overall complexity of computing this step (when ignoring the time of computing the new states) is $O(|O_p| r_{max} |S_A| w^{init}_{max})$.

In step 15, the coalition chooses the maximum value among the w_{max}^{init} values that have already been computed. Since this step is performed for each state $s \in S_s$, and for each $r \in \{0,...,r_{max}\}$, the overall complexity of performing this step is $O(w_{max}^{init}|S_A|r_{max})$. Computing the coalition's new state s' when it encounters opportunity $\vec{o}$ given that its current state is s can be done as described in the infinite decision horizon case. Thus, the overall complexity of computing future states for all states is $O(|S_A||A||O_p|)$ in B2C markets, and $O(|S_A||A|^3|O_p|)$ in C2C markets. Given this analysis, the overall complexity of the algorithm is $O(|S_A|\log|S_A| + |S_A||O_p|w_{max}^{init}r_{max} + |S_A||A|^2 + |S_A||A||O_p|)$ in B2C markets, and $O(|S_A|\log|S_A| + |S_A||O_p|w_{max}^{init}r_{max} + |S_A||A|^3|O_p|)$ in C2C markets. Hence the algorithm is polynomial in w_{max}^{init} and r_{max}.[16] $\square$

Similar to the infinite decision horizon a significant improvement in the above algorithm's performance can be achieved by calculating and using the specific upper bound, $w_{max}^{s_i}$, for each state, s_i in step 11, according to Equation 7.

In the following section we illustrate some properties of the PCS model both for the infinite and finite decision horizons model variants.

8. Illustrative Examples of the PCS Model

With an efficient means for calculating the coalition's optimal strategy when using parallel cooperative search, we can now demonstrate some specific properties of this search method. As a reference we use the Single Agent's Parallel Search (SAPS), the Single Agent's Sequential Search (SASS) and the Sequential Cooperative Search (SCS) models.

8.1 Infinite Decision Horizon

We begin by illustrating the parallel cooperative search with an infinite decision horizon. First we demonstrate the influence of the level of heterogeneity in the utility functions of the different coalition members on the coalition's performance (in terms of the expected utility achieved). In order to demonstrate this we use the following environment, which was used originally for evaluating the performance of the SCS model (Sarne & Kraus, 2005):

Environment 5. *A coalition of two agents, a_1 and a_2, searching for opportunities defined by two attributes, B_1 (e.g., quality) and B_2 (e.g., store rating), where each attribute can have a value from the discrete range of $(1,...,5)$ with an equal probability for each of the values. The agents are heterogeneous in respect to the way they evaluate each potential opportunity. Agent a_1 is associated with the utility function $U_1(\vec{o}) = B_1 + B_2$, while agent a_2 is associated with the utility function $U_2(\vec{o}) = 2(1-\alpha)B_1 + 2\alpha B_2$. Thus, the parameter α indicates the level of the agents' similarity/heterogeneity. The search cost of any single agent for conducting a single interaction is c_{base} for w parallel interactions ($w > 1$): $c(w,1) = c_{base} + c_{parallel} * (w-1)$. The search cost of a coalition of size n is $c(w,n) = c(w,1)ln(n+1)$, $\forall(n > 1)$.*

Figure 2 depicts the expected utility per agent when using the different search methods[17] in the C2C market (left hand-side) and the B2C market (right hand-side) as a parameter of the similarity level, α, between the utility functions of the agents constituting the coalition.

16. The algorithm uses w_{max}^{init} as an upper bound for the optimal number of interactions $x^*(s_i)$, $\forall i = 1,...,|S_A|$. This bound is valid since according to Equation 7 the value of $w_{max}^{s_i}$ increases as $V_t(s_i)$ decreases and the coalition's termination utility reaches its minimum in the initial state ($V_t(\{\}) = 0$).
17. For the cooperative models the average expected utility per coalition member measure is used.

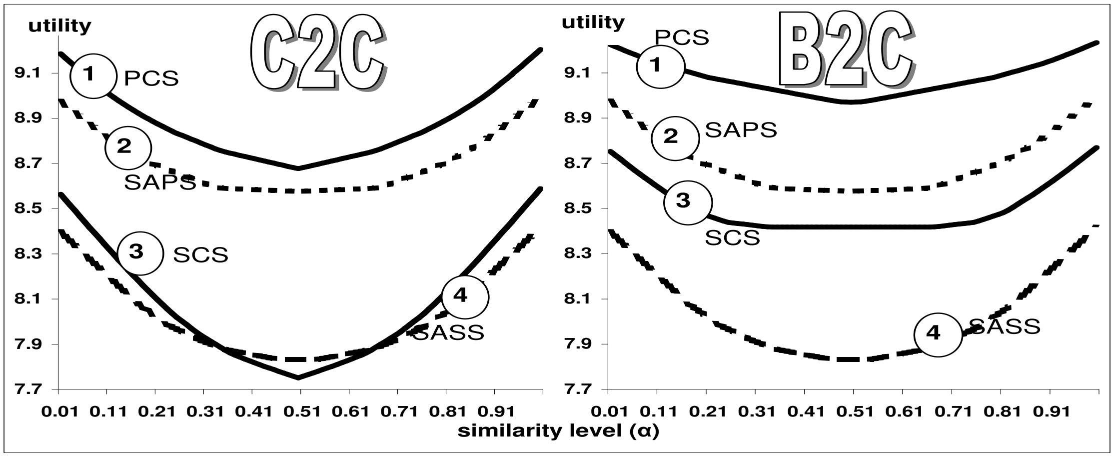


Figure 2: Average expected utility per buyer agent as a function of similarity level for different models in different markets

Curve 1 in each graph depicts the average expected utility when the two agents form a coalition as a function of the similarity level α between the agents' utility functions, making use of the suggested parallel cooperative search. Here, the search cost of conducting a single interaction was set at 0.2 ($c_{base} = 0.2$) and the search cost to conduct additional interactions was set to 0.05 ($c_{parallel} = 0.05$). As expected our model (represented by curve 1) outperforms the SCS model (represented by curve 3) in terms of the expected utility for the agents. The other two curves describe the average expected utility of the agents when each searches separately using the Single Agent Parallel Search (SAPS) (represented by curve 2) and the Single Agent Sequential Search (SASS) (represented by curve 4) models. In this specific environment the use of the cooperative parallel search also outperforms the single agent parallel search model though this is not always the case. Notice that the results obtained for the cooperative parallel search are consistent with a general characteristic of cooperative search (Sarne & Kraus, 2005) by which the use of the method in the B2C market results in a better expected utility than in the C2C market. In the case of the separated single searches (SAPS and SASS models) the market type does not affect the strategy structure nor the expected utility since each agent searches only for a single opportunity for its own benefit.

Figure 2 also reflects an interesting insight which contradicts an important strategy domination relationship found between single and cooperative sequential search techniques of fully homogeneous agents (i.e. with the same utility function, as in the case where $\alpha = 0.5$ in our example) operating in C2C markets. While for the sequential search the use of the single agent search always outperforms the cooperative search in the C2C markets (when considering fully homogeneous agents) (Sarne & Kraus, 2005), here we have actual evidence that where parallel search is concerned, the cooperative search technique may outperform the aggregated result of the single homogeneous agents' search.

Figure 3 shows the expected utility per agent when using the different search methods in the C2C market (left hand-side) and the B2C market (right hand-side) as a parameter of the cost of conducting an additional interaction $c_{parallel}$ (notice that the agent's performance is not affected by this value in the SASS and the SCS models). The results are based on Environment 5 (the same environment we used for Figure 2), where $\alpha = 0.1$ and $c_{base} = 0.2$. As expected as the cost $c_{parallel}$

decreases the average expected utility using the parallel models (PSC and SAPS models) increases (and thus the greater the improvement in comparison to the sequential models). Note that whenever $c_{parallel} \geq c_{base}$ the performance converges to the one achieved by obtaining one observation at a time, as used in the sequential models. This behavior is correlated with Proposition 3.

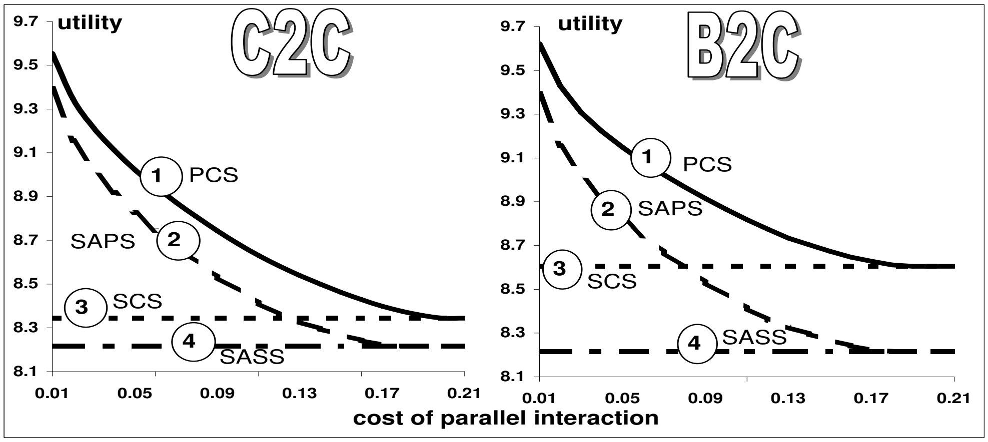


Figure 3: Average expected utility per buyer agent as a function of $c_{parallel}$ when using the different search methods in different markets

The cost of conducting an additional interaction $c_{parallel}$ also influences the optimal number of interactions that the coalition should conduct. Figure 4 shows the optimal number of parallel interactions that the coalition should conduct at the beginning of its search, $x^*(\{\})$, as a function of $c_{parallel}$ in the same environment that we used for Figure 3 (Environment 5). As expected, the optimal number of interactions, $x^*(\{\})$, increases as $c_{parallel}$ decreases and is equal to 1, when $c_{parallel} \geq c_{base}$.

In Figure 4 the coalition's optimal number of interactions using the PCS model is higher when the coalition operates in the C2C market than when in the B2C market. While this can have an intuitive explanation (in the C2C market each opportunity can be exploited by only one of the agents so the coalition needs to encounter more opportunities in the C2C market) it cannot be generalized. The following example illustrates a scenario in which the optimal number of interactions is actually greater when operating in a B2C market.

Environment 6. *A coalition of two agents, a_1 and a_2, searching for a product (e.g., a computer game) associated with 4 types of opportunities $\{\vec{o}_1, \vec{o}_2, \vec{o}_3, \vec{o}_4\}$ (e.g., representing different configurations). The agents' utilities and opportunities distribution are given in Table 5. The search cost of any single agent to conduct a single interaction is $c_{base} = 0.2$, and the search cost of conducting an additional search is $c_{parallel} = 0.1$, $c(w,1) = c_{base} + c_{parallel} * (w-1)$. The search cost for a coalition is $c(w,n) = c(w,1)ln(n+1)$, $\forall(n > 1)$.*

Figure 5 shows the optimal number of interactions used by the coalition (in the SCS model) or the agent (in the SAPS model) at the beginning of the search, when operating in Environment 6. As one can see the coalition's optimal number of interactions using the PCS model in the B2C

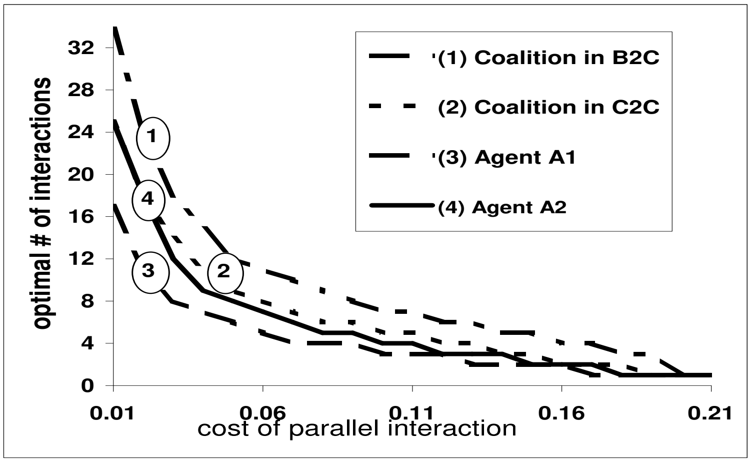


Figure 4: Coalition's or agent's optimal number of parallel interactions at the beginning of its search as a function of $c_{parallel}$

Opportunity	*Probability*	*Utility*	
		Agent a_1	*Agent a_2*
$\vec{o}_1$	0.499	0	0
$\vec{o}_2$	0.25	0	1
$\vec{o}_3$	0.25	1	0
$\vec{o}_4$	0.001	100	100

Table 5: Agents' utilities for the four opportunities in Environment 6

market (44) is larger than the coalition's optimal number of parallel interactions in the C2C market (8). Moreover this figure contradicts two additional hypothesis that one may presume about the optimal number of a coalition's parallel interactions. The first is that the coalition's optimal number of parallel interactions is at most equal to the overall number of parallel interactions, when each of the agents conducts the search autonomously. As can be seen in Figure 5, the optimal number of parallel interactions in the B2C market is greater than the total number of parallel interactions, when each of the agents conducts its search autonomously ($10+10=20<44$). The second hypothesis suggests that the coalition's optimal number of parallel interactions is at least the number of parallel interactions, when each of the agents conducts the search autonomously. This, again is proven wrong in Figure 5, where the coalition's optimal number of parallel interactions in the C2C market (8) is smaller than the number of parallel interactions of agent a_1 (10), and the number of parallel interactions of agent a_2 (10).

Next we introduce and make use of a simpler sample environment for demonstrating some additional properties of the cooperative parallel search.

Environment 7. *A coalition of two agents, a_1 and a_2, searching for opportunities defined by two attributes, B_1 (e.g., quality) and B_2 (e.g., store rating), where each attribute can have a value from the discrete range of $(1,2)$ with an equal probability for each of the values. The utility functions used are $U_1(\vec{o}) = 1.9B_1 + 0.1B_2$ and $U_2(\vec{o}) = 0.1B_1 + 1.9B_2$. The search cost of a single agent*

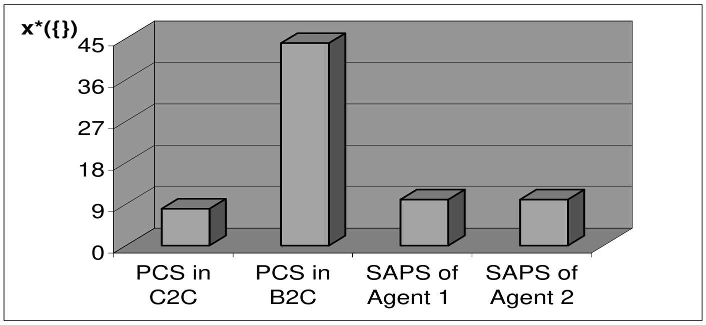


Figure 5: Coalition's optimal number of parallel interactions at the beginning of its search

is $c(w,1) = 0.5 + 0.05w$, *and for a coalition it is* $c(w,n) = c(w,1) * ln(n+1)$, $\forall(n > 1)$. *Table 6 summarizes the environment's setting.*

Opportunity	*(Attribute1, Attribute2)*	*Probability*	*Utility*	
			Agent a_1	*Agent* a_2
$\vec{o}_1$	(1,1)	$\frac{1}{4}$	2	2
$\vec{o}_2$	(1,2)	$\frac{1}{4}$	2.1	3.9
$\vec{o}_3$	(2,1)	$\frac{1}{4}$	3.9	2.1
$\vec{o}_4$	(2,2)	$\frac{1}{4}$	4	4

Table 6: Agents' utilities for the four opportunities in Environment 7

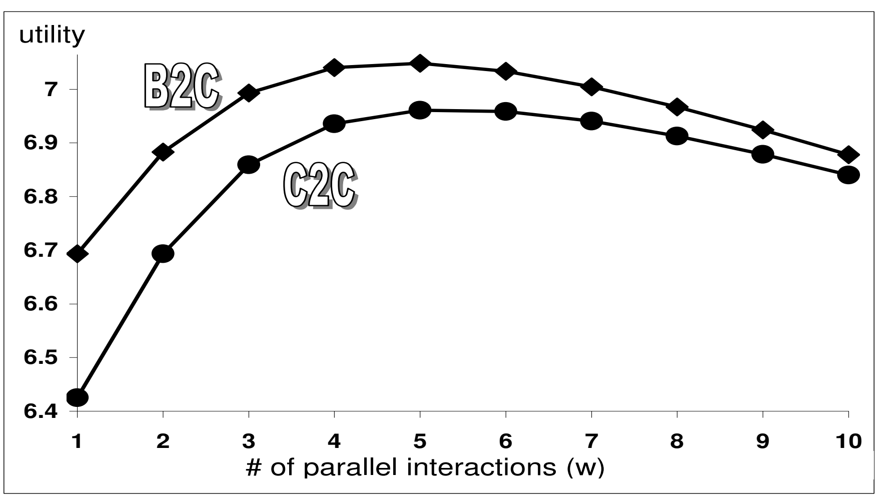


Figure 6: Coalition's overall expected utility as a function of *w*

Figure 6 depicts the expected coalition's overall utility with respect to the number of interactions conducted at the beginning of the search (i.e. the first search stage, before the coalition knows about any of the opportunities), assuming that in all the other states the coalition uses the optimal number of parallel interactions, $x^*(s)$. Here, we can see the effect of two conflicting forces: as the number of parallel interactions the coalition uses in this stage increases, the probability of associating a better opportunity with any of the two coalition members increases. However the overall search cost associated with the search stage increases. From the figure, we conclude that the optimal number of parallel interactions to be used in this stage is $x^*(\{\}) = 5$.

An additional important characteristic of the cooperative parallel search we wish to emphasize concerns the number of parallel interactions used as part of the optimal search strategy along the search. While in a single agent's parallel search the search strategy is stationary (i.e. the number of parallel interactions used does not change along the search process) in our model the number of parallel interactions along the search that needs to be maintained depends on the coalition's state (i.e. the set of opportunities known to the coalition). This is demonstrated in the directed acyclic graph (DAG) given in Figure 7, which describes the search process in Environment 7. The vertices of this graph present potential coalition's states (the state is determined according to the relevant set of known opportunities, correlated with the definition given in section 4). An edge connects two states s and s' only if there is a possibility to reach state s over the following search round if the coalition conducts the search according to the optimal search strategy. For example a directed edge connects between $s = \{\}$ and $s' = \{(1,2),(2,1)\}$, since the coalition can proceed from s to s' if it conducts the five parallel interactions according to the optimal search strategy and encounters opportunities $(1,2)$ and $(2,1)$. Notice that when reaching states $\{(1,2),(2,1)\}$ and $\{(2,2),(2,2)\}$ the optimal strategy of the coalition is to terminate the search. Therefore there is no edge originating in these states. As illustrated in Figure 7, the number of parallel interactions the coalition should use according to the optimal search strategy (denoted by $x^*(s)$) depends on the coalition's state. For comparison purposes, notice that in any of the single agents' separate search (i.e. the single agent parallel search model) the optimal strategy is to constantly use 4 parallel interactions (as long as the agent's strategy is to resume the search).

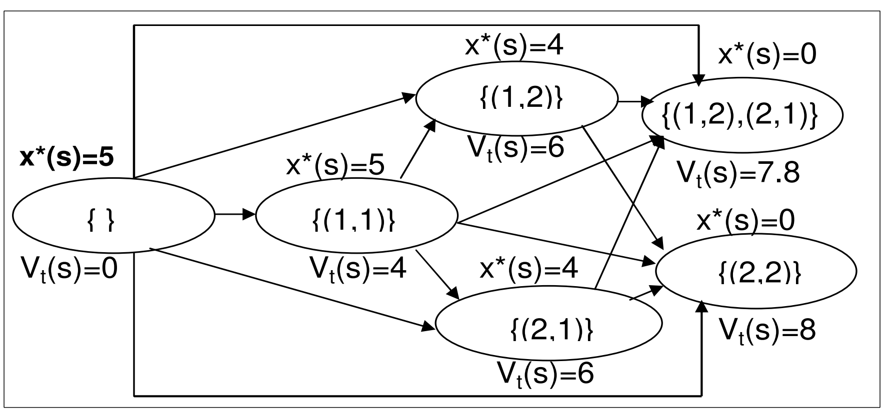


Figure 7: Optimal strategy and potential transitions between states in a simple B2C market

8.2 Finite Decision Horizon

In this section we demonstrate some properties of the PCS model variant where the coalition is given a finite decision horizon. First we explore the influences of the cost of conducting an additional interaction, $c_{parallel}$, on an agent's expected utility. Figure 8 depicts the results obtained from varying $c_{parallel}$ between 0.01 and 0.3. In this figure we use the same environment used for Figure 3 (Environment 5), where $\alpha = 0.1$, $c_{base} = 0.2$ with a decision horizon of two search rounds. As the cost of conducting an additional interaction $c_{parallel}$ decreases, the coalition's expected utility using the parallel models PSC and SAPS increases and its superiority over the sequential models increases. Moreover as shown in Figure 8 the parallel model outperforms the sequential model even when $c_{parallel} \geq c_{base}$ (e.g., for $c_{parallel} = 0.2$). Recall that when $c_{parallel} \geq c_{base}$ the parallel technique does not improve the expected utility when having an infinite decision horizon (as suggested in Proposition 3).

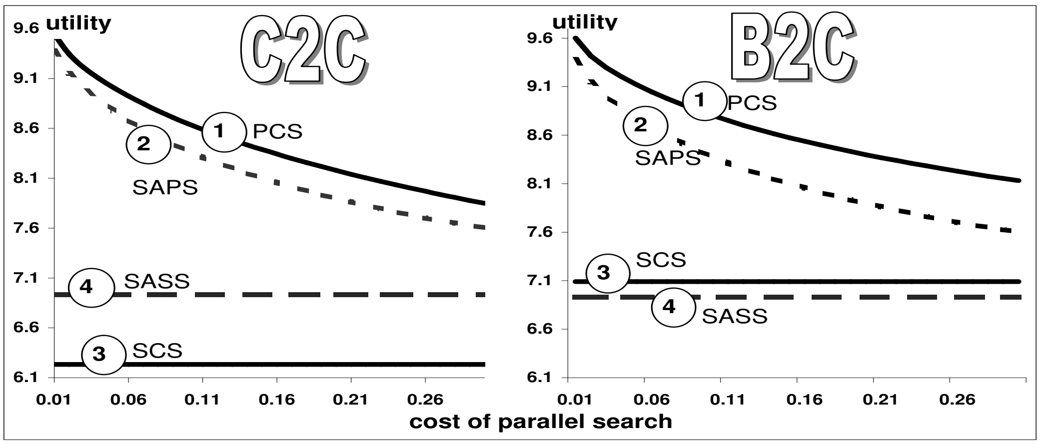


Figure 8: The average expected utility per buyer agent as a function of $c_{parallel}$ when the coalition should terminate its search within the two next rounds.

Another factor that affects the expected utility is the decision horizon, represented by the value of the parameter r. Figure 9 presents the coalition's expected utility as a function of r. In this figure we used the same environment that we used in Figure 8, where we set the cost of each additional search, $c_{parallel} = 0.2$. As observed in Figure 9, in all search models, the earlier the coalition should terminate its search the smaller its expected utility. Moreover, the expected utility improvement obtained in the PCS (in comparison to the SCS) and SAPS (in comparison to the SASS) models increases as r decreases. In this case the ability to conduct parallel interactions compensates for the small number of search rounds that the coalition can conduct. Note that as r increases, the average expected utility per buyer agent converges to the average expected utility per buyer agent in the infinite decision horizon model.

Finally, Figure 10 depicts the optimal number of interactions for a coalition (in the PCS model) or an agent (in the SAPS model) at the beginning of the search, as a function of r. For this figure we used the same environment parameters that were used in Figure 9. As expected the number of parallel interactions increases as r decreases.

As we have demonstrated throughout this section, the PCS outperforms the sequential cooperative search. Generally, the magnitude of improvement depends on the size of the domain, the search cost structure, and the different utility functions used.

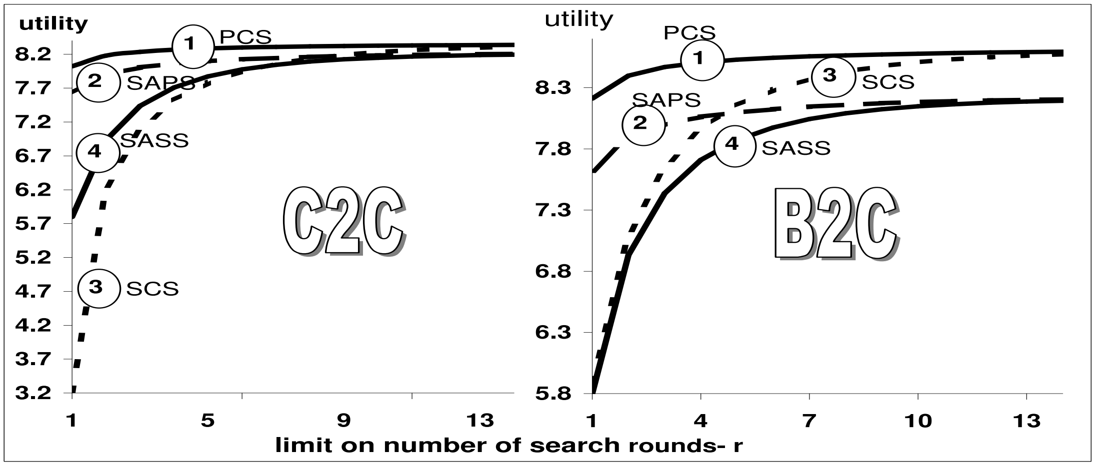


Figure 9: The average expected utility per buyer agent as a function of the limit of the number of search rounds, r

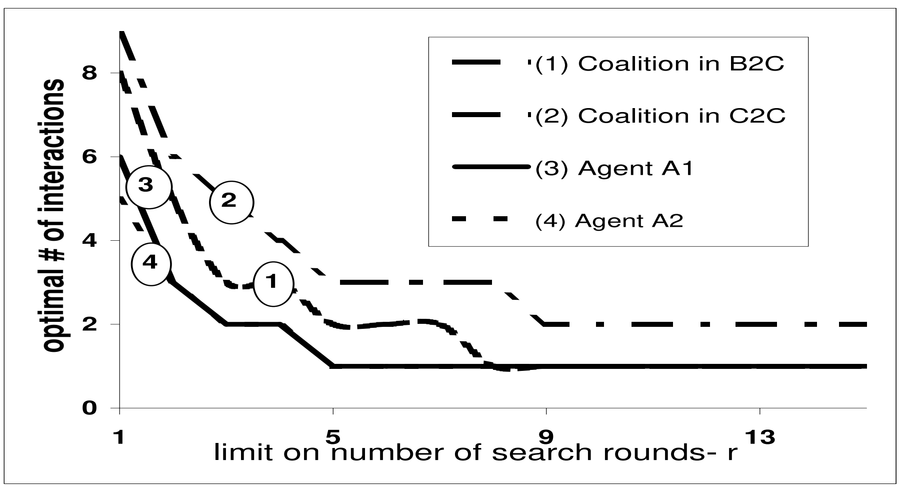


Figure 10: Optimal number of interactions as a function of the decision horizon, *r*

9. Discussion and Conclusions

The capability of using parallel interactions as part of a search process is inherent in the infrastructure of autonomous information agents. When using the cooperative parallel search, the coalition should use a new strategy, different in its structure in comparison to the optimal strategy used in the cooperative sequential search and in a single agent's parallel search. As expected, the use of the new model has the potential of significantly improving the coalition's expected utility as demonstrated in the previous section. Furthermore, we emphasize that the coalition's expected utility will never decrease when using our proposed mechanism in comparison to pure sequential cooperative search. This is mainly because the suggested algorithm will converge to one interaction at a time strategy, as used in the sequential cooperative search (which is a specific case of our model) in the case where maintaining more than a single interaction in some of the world states is not favorable. Obviously if the search cost is linear and depends solely on the number of interactions being maintained, then there is no use for the coalition to increase the number of sellers with whom it interacts in a given search round. Nevertheless, scenarios in which the coalition's search cost combines additional fixed components or non-linear dependency on the number of interactions maintained are much more realistic (Sarne & Kraus, 2005). In these scenarios the parallel cooperative search yields large benefits for searchers.

While the parallel cooperative search weakly dominates the sequential cooperative search, it does not necessarily dominate autonomous search (where the agents search by themselves instead of cooperatively). The decision of whether to use the parallel cooperative search autonomously depends on the amount of coalition overhead costs induced by the cooperative search. Therefore, having the computational means developed in this work enables the agents to identify fruitful opportunities for searching cooperatively. Generally, the introduction of the parallel cooperative search model substantially increases the number of scenarios in which agents will prefer to search cooperatively.

The novelty of the analysis given in this paper is threefold. First it supplies us with a better understanding of the space of opportunities, dividing it into improving and non-improving areas. Thus, instead of having dual simultaneous dependencies between states we can now define a single directional dependency for each pair of states. Second, it supplies a bound for the optimum number of parallel interactions that the coalition uses in each state in its optimal strategy. Third we represent the parallel search as a sequential process, without breaching any of the model assumptions. These three features allow us to overcome the main complexity associated with the attempt to solve the problem as a set of equations and proffer a finite algorithm that is polynomial in the number of parallel interactions (rather than the brute-force algorithm which is exponential in the number of parallel interactions) and inevitably reaches the optimal strategy. Moreover, we provide a comprehensive analysis of the parallel model for extracting the optimal search strategy given a finite decision horizon. Here, as we illustrate, the coalition can benefit significantly more than the infinite decision horizon from the integration of parallel interactions into the cooperative search. The PCS model (both in its finite and infinite decision horizon forms) is general and can be applied to any coalition regardless of the cost function and the preferences used by its agents. Further adaptation to additional markets (other than C2C and B2C) can be achieved by appropriately modifying the allocation function used for assigning each agent with one of the opportunities found along the search.

The focus of this research is on finding the optimal search strategy for the coalition, given its structure, the opportunity distribution, and the reported members' preferences. It treats the coalition as a unified entity sharing a common goal (to maximize the sum of its members' utilities). Nevertheless, there are various, important aspects of coalition formation, in the context of cooperative search, that do not always correlate with the assumptions used and should therefore be further addressed. For example, there may be an incentive for coalition members to misreport their preferences when side-payments are used. Alternatively, agents may be able to form "side coalitions", and free-ride actively searching coalitions by having only a single agent as a member of the actively searching coalition (that member would act as a "spy" for the "side coalition"). Moreover, there can be environments where agents may face tight budget constraints that could be violated within coalition side-payments. The analysis of these important issues is based on the ability to properly derive the coalition's utility given any specific setting. This paper supplies this functionality, laying the foundation and enabling such research. Future work should encompass and extend the scope of research to include these additional topics associated with the coalition formation process, such as the coalition stability and the division of payoffs between the coalition members. Other important extensions should include the relaxation of some assumptions in the underlying opportunity model (e.g., different deadlines on different opportunities).

Acknowledgments

This work was supported in part by NSF grant no IIS0705587 and ISF. Kraus is also affiliated with UMIACS.

Appendix A. A Summary of Notations

Notation	Meaning
$B = (B_1, B_2, ..., B_k)$	The set of the attributes defining each of the potentially available opportunities in the market, where each attribute B_i can be assigned a value from the finite set $(b^i_{min}, ..., b^i_{max})$.
O_p	The space of potential opportunity types the coalition may encounter.
A	A coalition of agents.
$U_j(\vec{o})$	Agent a_j's utility from opportunity type $\vec{o}$.
$c(w,n)$	The search cost associated with having a coalition of n agents maintaining w simultaneous interactions with seller agents.
$alloc(\theta_{known})$	A function that maps a given set of opportunities θ_{known} to the coalition members in A in a way that the aggregated agents' utility is maximized
$state(\theta_{known})$	The state s of a coalition A acquainted with a set θ_{known} of known opportunities.
S_A	The set of all possible states of a coalition A throughout its search.
$V_t(\theta_{known})$:	The immediate utility for the coalition if it terminates the search given a set of known opportunities θ_{known}.
$V(s,w)$	The coalition's expected utility when using w parallel interactions when in state s.
$V(s,0)$	The immediate utility obtained, if the coalition decides to terminate the search at state s.
$x^*(s)$	The number of parallel interactions that the coalition conducts at state s according to its optimal strategy.
Θ_w	The collection of all w-sized sets of opportunities that can be produced in the environment the coalition operates.

Notation	Meaning
$p_w(\theta_w)$	The probability of encountering a specific set of opportunities θ_w, when maintaining w random interactions with seller agents.
$\Theta^s_{improve_w}$	The collection of all w-sized sets of opportunities, θ_w, that change the coalition's current state s.
$\Theta^s_{stay_w}$	The collection of all w-sized sets of opportunities, θ_w, that does not change the coalition's current state s.
S_s	The set of all states belonging to S_A, sorted according to their termination utilities $V_t(s)$.
$p^{stay}(s,w)$	The probability the agent stays at the same state s after conducting w parallel interactions.
w^s_{max}	The upper bound for the optimal number of parallel interactions to be used when the coalition in state s.
w^{init}_{max}	The upper bound for the optimal number of parallel interactions to be used when the coalition begins its search $(s = \{\})$.
$V^{new}(s,k)$	The coalition's expected utility obtained by potentially reaching new states (e.g. different than s) after executing k parallel interactions (without incorporating the search cost of conducting these k interactions), assuming future strategy uses $x^*(s')$ in each new future state s'.
Finite Decision Horizon	
r	The maximum number of search rounds that the coalition can conduct.
$x^*(s,r)$	The coalition's optimal strategy at state s, if it needs to terminate its search within the next r search rounds.
$V^*(s,r)$	The expected utility of a coalition restricted to maximum r further search rounds, when reaching a state s assuming it acts optimally throughout its search.
$V(s,w,r)$	The coalition's expected utility when it conducts w interactions in the next round and has to terminate its search within the next r rounds.

References

Avis, D., & Lai, C. (1988). The probabilistic analysis of a heuristic for the assignment problem. *SIAM J. Comput.*, *17*(4), 732–741.

Bakos, Y. (1997). Reducing buyer search costs: Implications for electronic marketplaces. *Management Science*, *42*(12), 1676–92.

Baron, D. P., & Ferejohn, J. A. (1989). Bargaining in legislatures. *American Political Science Review*, *83*(4), 1181–1206.

Benhabib, J., & Bull, C. (1983). Job search: The choice of intensity. *J. of Political Economy*, *91*(5), 747–764.

Breban, S., & Vassileva, J. (2001). Long-term coalitions for the electronic marketplace. In *B. Spencer, ed., Proceedings of E-Commerce Applications Workshop*.

Burdett, K., & Malueg, D. A. (1981). The theory of search for several goods. *Journal of Economic Theory*, *24*, 362–376.

Carlson, J. A., & McAfee, R. P. (1984). Joint search for several goods. *Journal of Economic Theory*, *32*, 337–345.

Choi, S., & Liu, J. (2000). Optimal time-constrained trading strategies for autonomous agents. In *Proceedings of MAMA-2000*, pp. 11–13.

Dias, M. (2004). *TraderBots: A New Paradigm for Robust and Efficient Multirobot Coordination in Dynamic Environments*. Ph.D. thesis, Robotics Institute, Carnegie Mellon University.

Gal, S., Landsberger, M., & Levykson, B. (1981). A compound strategy for search in the labor market. *International Economic Review*, *22*(3), 597–608.

Gatti, J. (1999). Multi-commodity consumer search. *Journal of Economic Theory*, *86*(2), 219–244.

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems, Science and Cybernetics*, *4*(2), 100–107.

Ito, T., Ochi, H., & Shintani, T. (2002). A group-buy protocol based on coalition formation for agent-mediated e-commerce. *International Journal of Computer and Information Science (IJCIS)*, *3(1)*, 11–20.

Kahan, J., & Rapoport, A. (1984). *Theories of Coalition Formation*. Hillsdale, NJ:Lawrence Erlbaum Associates.

Keeney, R., & Raiffa, H. (1976). *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. New York, US:John Wiley & Sons.

Kephart, J., & Greenwald, A. (2002). Shopbot economics. *JAAMAS*, *5*(3), 255–287.

Lermann, K., & Shehory, O. (2000). Coalition formation for large scale electronic markets. In *Proceedings of ICMAS-00*, pp. 167–174.

Li, C., Rajan, U., Chawla, S., & Sycara, K. (2003). Mechanisms for coalition formation and cost sharing in an electronic marketplace. In *Proceedings of ICEC-03*, pp. 68 – 77.

Lippman, S., & McCall, J. (1976). The economics of job search: A survey. *Economic Inquiry*, *14*(3), 155–189.

Manisterski, E. (2007). *Protocols and Strategies for Agents Teamwork*. Ph.D. thesis, Department of Computing Science, Bar Ilan University.

McMillan, J., & Rothschild, M. (1994). Search. In Aumann, R. J., & Sergiu Hart, A. (Eds.), *Handbook of Game Theory with Economic Applications*, pp. 905–927. Elsevier.

Morgan, P. (1983). Search and optimal sample size. *Review of Economic Studies*, *50*(4), 659–675.

Morgan, P., & Manning, R. (1985). Optimal search. *Econometrica*, *53*(4), 923–944.

Rothschild, M. (1974). Searching for the lowest price when the distribution of prices is unknown. *Journal of Political Economy*, *82*(4), 689–711.

Sandholm, T., Larson, K., Andersson, M., Shehory, O., & Tohme, F. (1999). Coalition structure generation with worst case guarantees. *Artificial Intelligence*, *111*(1-2), 209–238.

Sarne, D., & Kraus, S. (2003). The search for coalition formation in costly environments. In *Proceedings of CIA-03*, pp. 117–136.

Sarne, D., & Kraus, S. (2005). Cooperative exploration in the electronic marketplace. In *Proceedings of AAAI-05*, pp. 158–163.

Shehory, O., & Kraus, S. (1998). Methods for task allocation via agent coalition formation. *Artificial Intelligence*, *101*(1-2), 165–200.

Stigler, G. (1961). The economics of information. *Journal of Political Economy*, *69*(3), 213–225.

Talukdar, S., Baerentzen, L., Gove, A., & de Souza, P. S. (1998). Asynchronous teams: Cooperation schemes for autonomous agents. *Journal of Heuristics*, *4*(4), 295–321.

Tsvetovat, N., Sycara, K., Chen, Y., & Ying, J. (2000). Customer coalitions in electronic markets. In *Proceedings of AMEC-00*, pp. 121–138.

Wang, Y., Makedon, F., & Ford, J. (2004). A bipartite graph matching framework for finding correspondences between structural elements in two proteins. In *Proceedings of EMBC-04*, Vol. 42, pp. 2972–75.

Yamamoto, J., & Sycara, K. (2001). A stable and efficient buyer coalition formation scheme for e-marketplaces. In *Proceedings of Agents-01*, pp. 576–583.