A Procedure for the Allocation of Two-Dimensional Cylindrical-Shaped Resources in a Multiagent System

Karthik Iyer IYERK@ENGR.SC.EDU
Michael N. Huhns HUHNS@ENGR.SC.EDU
Department of Computer Science and Engineering
University of South Carolina
Columbia, SC 29208 USA

Abstract

This paper presents protocols and procedures for allocation of resources that have two dimensions and can be characterized by a cylindrical topology. A resource has the topology of a cylindrical surface when one of its dimensions is finite and bounded (the axial dimension) and the other is finite and unbounded (herein meaning it does not have a distinct boundary). Real world examples show the practical significance of the problem. To solve it, we first show how the cylindrical resource allocation problem can be mapped into a one-dimensional circular resource allocation problem for which we previously obtained a solution. Second, we describe a more flexible cylindrical resource allocation protocol wherein agents mark *n* non-overlapping rectangles of equal value (by their valuation) on the cylindrical surface of a resource. This requires that two conditions, A and B, be considered. If condition A is true, then the existing allocation problem is transformed to a one-dimensional linear resource allocation problem and results from our earlier work are applicable. Condition B uses the notion of degree of partial overlap to create a sufficiency condition for the existence of a solution. Our more general two-dimensional cylindrical resource allocation procedure is then used to find the solution. The inter-relationship of conditions A and B is discussed. Our procedure is fair, strategy-proof, constructive, and does not need the resource to be measurable. We end with a discussion of improvements that can be made to the procedure and extensions of this work to other domains and topologies.

1. Introduction

Multiagent systems (MAS) have become an important paradigm to solve problems in distributed computing. MAS is being used to model and solve various real world problems such as operating distributed sensor networks (Soh & Tsatsoulis, Nov 2005), facilitate ecommerce (He & Leung, 2002) and allocate scarce resources (Bredin et al., June 2000). An important feature of MAS is the expectation of non-cooperative (i.e., competitive) behavior among agents. Agents are assumed to be selfish, autonomous (no global control) and capable of lying to gain the maximum possible utility. The negotiation protocols that are designed to enable agent interaction are therefore expected to be resistant to such manipulative behavior by agents. MAS researchers have suggested criteria (Rosenschein & Zlotkin, 1994) that negotiation protocols should fulfill in order to guarantee that the resource allocation is satisfactory to all agents. Stability, simplicity and strategy-proofness are examples of such criteria. A detailed list of the criteria for negotiation and resource allocation has been presented in our earlier work (Iyer & Huhns, Feb 2007a). Once the agents have completed their interactions based on the protocol, a follow-up procedure may be needed to apportion the resource. Researchers in MAS have generally assumed that the resource allocation is completed at the end of the negotiation stage or a simple degenerate procedure is used to finish the allocation process. Allocation procedures are therefore never explicitly discussed. Considerable contributions on allocation procedures have been made by mathematicians, who have studied the problem of resource allocation as early as 1948 (Steinhaus, 1948). The resource is modeled as a cake that needs to be apportioned to various people and hence the label "cake-cutting procedure." There exist numerous cake-cutting procedures that fulfill a variety of "nice" criteria that would enable a satisfactory allocation for participants. Criteria relevant to procedures include fairness, envy-freeness, and efficiency. (Brams & Taylor, 1996). A procedure is *fair* if every agent believes that it has received at least $1/n$ of the total resource allocated. A procedure is *envy-free* if every agent believes that the portion it has received is at least as large as the largest piece allotted to any agent. Efficiency is meant in the pareto optimal sense. An allocation is *pareto optimal* if there exists no other allocation of the resource such that at least one agent is better off while the others do no worse. The ideas in cake cutting are quite relevant to MAS, because mathematicians have made assumptions about the participants (people) quite similar to the ones in MAS, viz.

1. People (agents) are selfish.
2. People (agents) are autonomous.
3. People (agents) can misrepresent their true preferences (i.e., they can lie).

In this paper, we present one-shot negotiation protocols that allow agents to reveal their preferences for various portions of the resource. This is less common than the typical iterative forms of negotiation. At the end of the negotiation, one of the agents volunteers to act as a mediator and executes the procedure. Based on the computation of agent preferences, there is one of two outcomes:

1. The procedure is able to find a solution and all agents get a fair deal.
2. The procedure is unable to find a solution and all agents receive the conflict deal, i.e., no agent receives any part of the resource.

Then follow-up procedures being proposed compute an allocation of the resources based on the revealed preferences of agents. We contend that the procedures are fair, strategy-proof, and verifiable. A procedure is said to be *verifiable* if the allocation of the resource is invariant to the bias of the mediator. The procedures are novel because they take into account the topology and dimensionality of the resource. Specifically, a procedure has been designed to allocate cylindrical shaped resources. We explain the significance of this issue in the next section. This paper extends earlier work done in this domain by accounting for resources with higher dimensionality and a different topology. Solutions already exist for one-dimensional resource allocation for $n \leq 3$ agents (Huhns & Malhotra, July 1999; Stewart, December 1998) and for general n number of agents (Iyer & Huhns, Oct 2005). A qualified solution (Iyer & Huhns, Feb 2007b) also exists for allocating on a two-dimensional planar resource for a general n number of agents. The objective of the current paper is to propose negotiation protocols and procedures for allocation of two-dimensional cylindrical resources. The next section provides the motivation for such problems. The literature survey mentions the current state of research on resource allocation and negotiation. Then, we present the various versions of allocation procedures and protocols that transform the two-dimensional resource allocation problem into a one-dimensional resource allocation problem and use existing one-dimensional procedures to obtain an allocation. In order to do this, we check if the agent preferences satisfy certain qualifying conditions. Next, we specify a native two-dimensional allocation procedure that can guarantee an allocation provided specific conditions are fulfilled. This is followed by a discussion of the various issues involved with allocation of cylindrical shaped resources. Finally, we state our conclusions and suggest future work.

1.1 Motivation

An example will illustrate the idea of a cylindrical-shaped resource. Current FCC auctions (Goeree, Holt, & Ledyard, July 2006) for allocation of radio frequencies consists of allocating a particular band to the highest bidder. The frequency so allocated is then available to the winning agent for full-time use. Say, Acme cellular phone service wins the bid for a frequency band in the 1.0-1.2 GHz range. Due to the nature of the auction, even if the company only intends to use the allotted band from 7am to 11pm (peak usage hours); it needs to reserve the frequency for all 24 hours of the day. Another company Biffco Inc. wants to provide data streaming and downloading services to handheld devices during off-peak hours. Then it too has to bid for and reserve frequency bands for the entire 24 hours, which would be a more expensive (and potentially financially unviable) proposition. What is required is an auction mechanism where it is possible to bid for frequency bands for specific times of the day. Such an allocation makes more efficient use of the resource (i.e., the frequency spectrum). Such a modified auction would have a 24 hour clock as one dimension and frequency spectrum as the other dimension. Biffco's bid for off-peak hour usage of a frequency band is shown by the shaded rectangle in Figure 1.

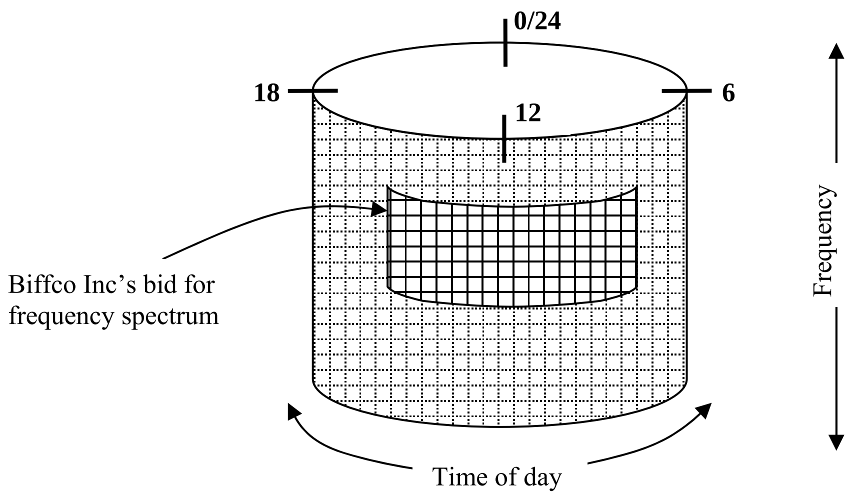


Figure 1: Bidding for frequency bands in the modified FCC auction

Another example would be a sports stadium where seats can be auctioned off for an upcoming football match. The schematic of the sports stadium is shown in Figure 2. Various fan groups can bid for specific blocks of seats as shown by the shaded region in the figure. Since the topology of the stadium is the same as a cylinder, an allocation protocol and procedure that takes this shape into account would yield a more satisfactory result.

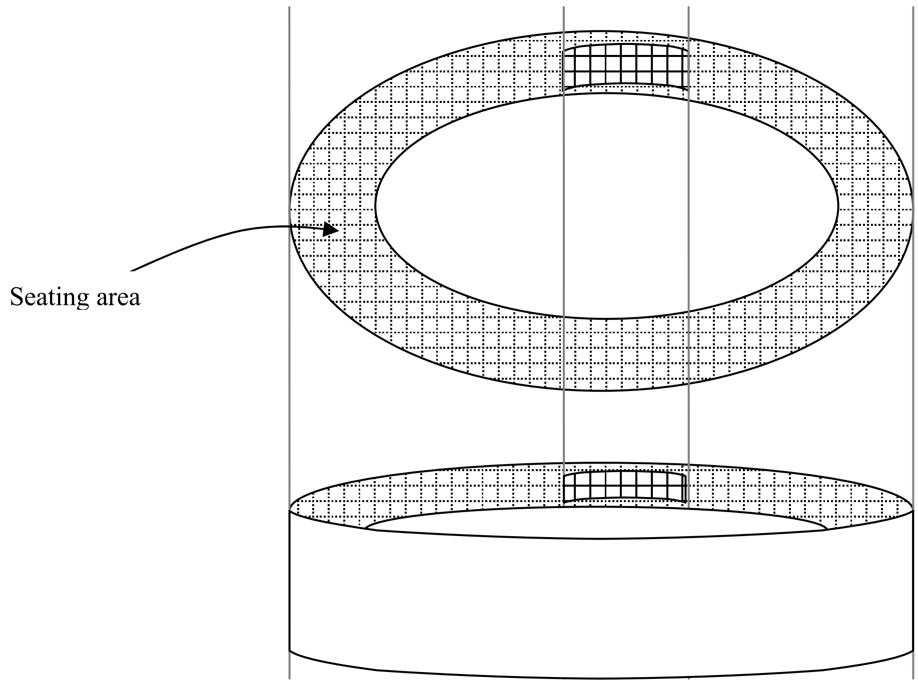


Figure 2: Bidding for seating blocks in a sports stadium

Can we quickly determine if a particular resource is cylindrical-shaped? Yes. A resource is said to be *cylindrical-shaped* if it is completely determined by exactly two independent finite variables, such that one of the variables is bounded while the other is unbounded. “Bounded” in this context means if the variable has distinguishable start and end points (i.e., one can identify distinct boundaries). Thus a line is a representation of a bounded variable. “Unbounded” means there exist no distinguishable start and end points. A circle (or any closed loop curve) is an example of an unbounded variable.

1.2 Topology of a Cylindrical Surface

In order to understand the applicability of the proposed procedure to various resource surfaces, the idea of a “cylindrical surface” needs to be explained. A cylindrical surface is obtained by gluing together a pair of opposing edges of a rectangle (Henle, 1994) Figure 3 illustrates the transformation of a rectangle to a cylinder.

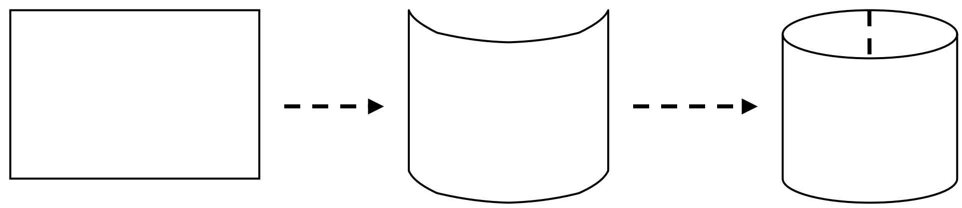


Figure 3: Transformation a two-dimensional rectangle to a cylindrical surface.

A cylindrical surface will have two surfaces and two edges. Shapes with very different geometry can have the same topology. A flat disc with a hole cut in the middle is actually topologically equivalent to the cylinder surface because it has the same number of edges and surface as that of a cylinder. The procedure and protocol that is proposed in this paper is *not yet* applicable to any shape of the cylindrical surface. Hence we specify the constrained set of shapes to which the procedures are applicable. The procedure is applicable to any cylindrical surface generated with the following parameters (qualified by constraints):

1. Shape of the generating curve: The generating curve should be closed loop and planar. Thus circles, ellipses or any arbitrary amoeba shapes on a two-dimensional plane qualify. The shape of the generating curve should not change as it moves along the axis, although the size can change.
2. Shape of the central axis: This is the line along which the curve moves to generate the cylindrical surface. The axis should be of finite length and can be any of any arbitrary shape in three dimensions as long as it does not form a closed loop.
3. The orientation of the generating curve with respect to the axis: The plane of the generating curve can be oriented at any *constant* angle with respect to the axis. However it cannot be parallel to the axis (as this would not generate a cylinder). Thus it is required that the angle of orientation be a non-zero finite value.
4. The scaling function for the generating curve: This parameter determines if the generating curve expands or contracts as it moves along the axis. This is used to generate cone and pot shaped surfaces. The scaling function can vary along the length of the axis, but it should always have a non-zero finite value.

Thus a cylindrical surface is generated by moving a planar closed-loop curve (of arbitrary shape) - oriented at a certain fixed angle with respect to the axis – along the axis (which is not a closed loop) while varying in size according to the scaling function.

The parameters and their respective constraints serve to describe the widest variety of shapes that are applicable for the procedure while ensuring that topological limitations are respected. Most of the standard cylindrical shapes that are encountered in real life are covered by the above description. Figure 4 shows some examples of well known cylindrical shapes along with the values for parameters needed to generate them. Note that the description still leaves out some shapes that have the topology of a cylinder. These are shapes generated by the planar curve that have varying angle of rotation as it moves along the length of the axis. The proposed procedure does not apply for such shapes.

The protocol to be put forth in the next section requires agents to draw rectangles on the cylindrical surface. How does one draw a rectangle on applicable cylindrical surfaces? We follow the analogy of drawing a two-dimensional planar rectangle where each pair of opposing sides is parallel to the X-axis and Y-axis respectively. In case of a cylindrical surface, one pair of sides is drawn parallel to the central axis (which need not be a straight line). The other pair of edges is drawn collinear to the generating curve at that point of the axis. Once rectangles are drawn in this manner (Figure 5), the proposed procedure can be applied to find feasible allocations. For ease of visualization, however, we will explain the details of the protocol and procedure using the standard geometric cylinder as a stand-in for all the applicable types of cylindrical surfaces.

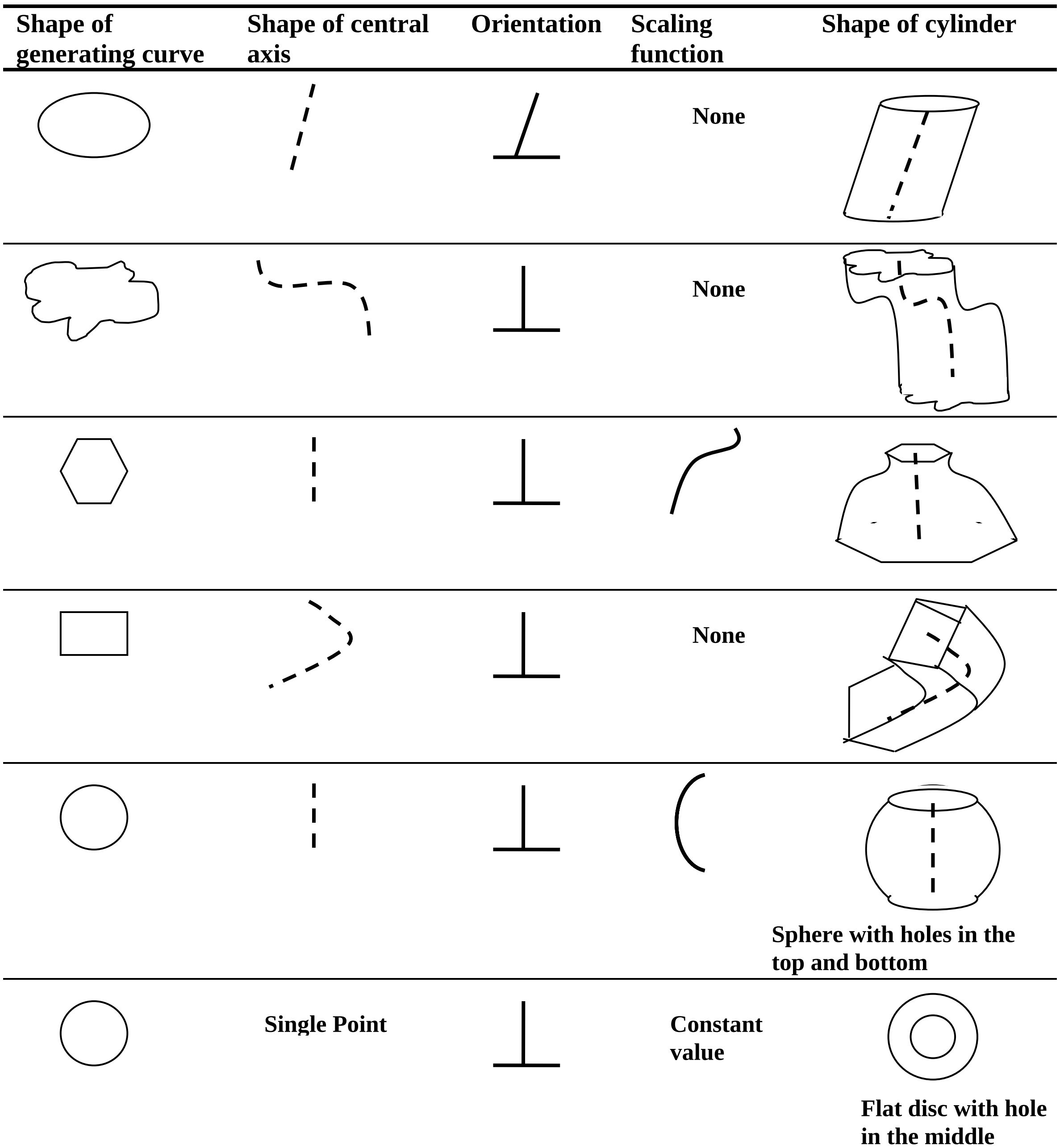


Figure 4: Generating cylindrical-shaped surfaces based on the various values of parameters

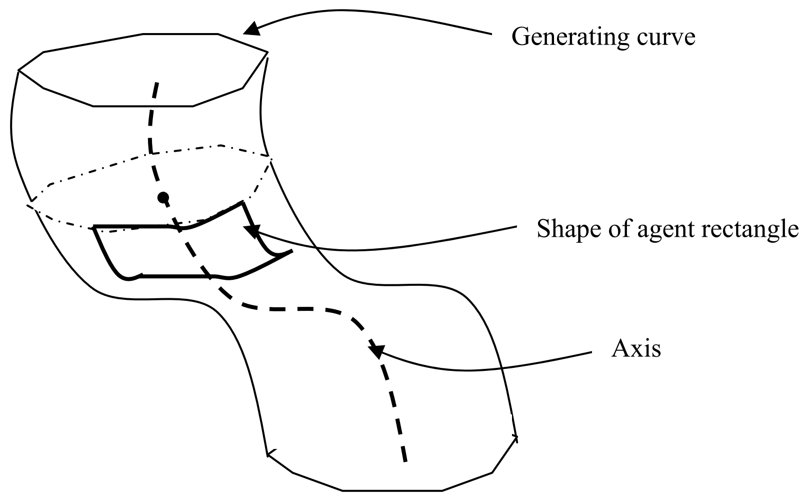


Figure 5: Drawing a rectangle on a cylindrical surface

A protocol and procedure for the allocation of a two-dimensional planar resource has been described (Iyer & Huhns, Feb 2007b). In this paper, protocols and procedures are proposed for the allocation of two-dimensional cylindrical resources that applies some of the ideas of the earlier work. The topological difference between a planar and cylindrical resource is shown in Figure 6.

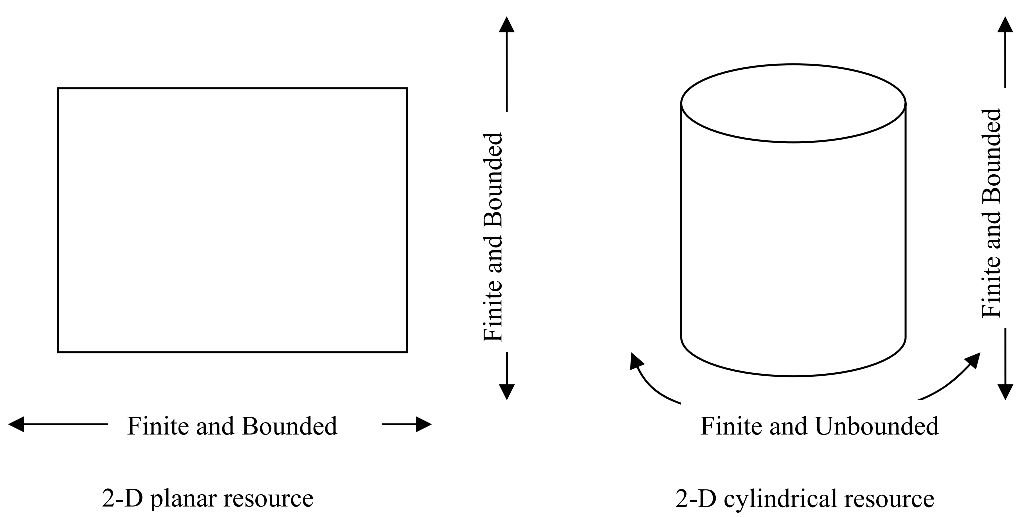


Figure 6: Topologies of planar and cylindrical resources

It is important to note that only the topology of the surface is important, not the geometry. Thus any surface having the topology of the cylinder is part of the domain of the proposed allocation protocol and procedure. We model the resources as a standard geometric cylinder simply because it is easier to illustrate the concepts, but the results apply to any shape that can be obtained from the continuous transformation of the cylindrical surface.

2. Background

The literature survey shows that there has been no explicit discussion on the allocation of cylindrical-shaped resources. This is because there are some implicit assumptions made about the "shape" of the resource, i.e., the resources have been assumed to be one dimensional and open-ended. The resource to be divided is commonly taken in the shape of a rectangular cake, wherein a knife is held parallel to one of the edges and cuts portions of the cake to allocate to agents. We classify existing work in this field into two categories: the one-dimensional cake-cutting problem and the two dimensional land division problem.

2.1 One-Dimensional Resource Sharing

The cake-cutting problem is a well known example of resource sharing among rational agents. Most of the literature that exists on the one-dimensional cake-cutting problem has been contributed by mathematicians. The classical solution for the two-person case, divide-and-choose, was first proposed by Steinhaus (1948). This solution, where one person divides a cake into two pieces and the other gets first choice of a piece, is both fair and envy free. However, it has been difficult to scale the solution to n agents. One of the solutions (Austin, Oct 1982) for dividing the cake among n agents fairly has been to use a moving knife parallel to one of the edges of the cake. The knife cuts when one of the agents yells “Cut!” and the portion traversed by the knife so far is allotted to the agent. This is an elegant and clean n agent solution for creating n portions fairly in $n-1$ cuts. But the moving knife solution has its own set of drawbacks. First, it is not envy-free, because an agent might evaluate all of the pieces—distributed after it was allocated a piece—and decide that one or more were larger than the piece it got. Second, it requires the presence of an unbiased external mediator who holds the knife and moves it along the cake at a constant rate. Despite this safeguard, it is difficult to verify the cutting of the cake. For example, if one of the agents alleges that the mediator cut the cake an inch shorter than it had expected, it would be difficult to find out who is telling the truth. In a distributed system, synchronization problems may also occur. An agent with a slower connection to the mediating authority will find its bid to cut may arrive later than that of an agent with a faster connection and may consequently lose the bid (the moving knife is similar to the open-cry descending-bid Dutch auction). Third, the moving-knife protocol is also not pareto optimal. A scheme (Sen & Biswas, 2000) to improve efficiency in the pareto optimal sense has been proposed with the use of two moving knives. However, the solution works only for division of the resource between two agents, and the agent that moves the knives must be able to estimate the utility function of the other agent well.

Other solutions to n-person division attempt to create a protocol that does not require assistance from an outsider. One way is to scale up from the two-person solution and iteratively add new agents until all have allocations. For $n=2$, the classic divide-and-choose is used. When agent 3 is added, agents 1 and 2 each divide their portions into three parts. Agent 3 then picks one part from each of the other agents. This continues as each agent is added. The drawback for this protocol is that the earlier agents will be faced with the chore of repeatedly dividing their portion into many pieces. The agent that is added last will get its share by doing the least amount of work.

Another solution (Tasnádi, 2003) converts the n-agent division problem into many $n-1$ agent problems and then recurses. The recursive calls return when the many two-agent problems are resolved and the answers back up to the top-level call. The drawback is that an agent whose shares remain unallocated till the end has to continuously re-bid for scattered pieces until the iterations end.

A divide-and-conquer procedure (Robertson & Webb, 1998) instructs the agents to cut the cake into half according to their measure. Then the cuts are ordered and the first $n/2$ cuts are allotted the left half of the cake. The rest are allotted the right half. This procedure continues until two agents have to cut the cake where the well known divide-and-choose algorithm can be implemented. An obvious drawback of this procedure is that the number of agents needs to be a power of two, which is an unrealistic requirement.

Stewart (December 1998) and Huhns and Malhotra (July 1999) discuss how to divide a strip of property along a coastline. However this is a specific solution applicable to only three agents. The result was further extended by the authors to be applicable to *n* agents (Iyer & Huhns, Oct 2005). The negotiation protocol described by these papers has useful features like: absence of synchronization problems, a mediator is not required and utility functions are not needed for the allocation procedure. However all of them assume the resource to be one-dimensional. In this paper we extend some of the ideas to allocate a two-dimensional cylindrical resource among agents.

Note that by specifying a restrictive protocol, the agents can be forced to reveal preferences so we can "downgrade" the two-dimensional cylindrical resource allocation problem into one-dimensional linear cake cutting problem. The various methods put forth so far in the literature can then be applied. However such methods will be less efficient in allocating resources and less flexible in allowing agents to specify their preferences in two-dimensions.

2.2 Two-Dimensional Resource Sharing

Two-dimensional resource sharing is generally mentioned as the "land division problem" in existing literature. Less frequently, the resources have been modeled as cakes and pizzas that need to be shared among people. The problems that have been framed for allocating such resources are however qualitatively different from the one that is being mentioned here. We have not come across any constructive (algorithmic) solutions to the cylindrical resource allocation problem. All papers have offered existential solutions to qualified versions of the land division problem, wherein the land is modeled as a two-dimensional planar resource. Hence existing literature is only tangentially relevant to the ideas proposed in this paper.

Hill (Oct 1982) was one of the first to tackle fairness issues in land division. The problem domain was qualified because it attempted to allocate portions of land to countries that shared a border with it such that each country received a portion connected with itself. He extends a non-constructive result proposed by Dubins and Spanier (Jan 1961).

The problem with Dubins and Spanier (Jan 1961) was that although it could create fair shares for all agents, the pieces might not be in the neighborhood of the country to whom the portion is allocated. Hill's solution is to create strips of land, thin enough, which connect the isolated portions, to the country it is allotted for. The strips created in this manner do not intersect because any pair of points in the set is assumed to be path-connected. There are some drawbacks to this approach. It can happen in reality that the strips that get created may be exceedingly thin so as to render the solution useless. For example, consider two countries that contest the land bordering them. It may be unacceptable for one country to have a single road connecting to its allocated portion surrounded by the enemy portions. Besides, this result does not explicitly provide a procedure for enabling such an allocation.

Beck (Feb 1987) proposes a semi-constructive result that improves upon Hill's paper. In order to do this he constructs a unit disc $|z|<1$ that is a homeomorphic map of D, the disputed territory. It is also assumed that individual circles and radial lines in the disc have zero measure. Next, various agents place "bids" in an "auction" submitting the smallest radius that will enclose a disc, which is valued at $1/n$ for that particular agent. The smallest such radius among the various bids is picked and the disc is awarded to that agent. In order that the portion may be connected to the agent's territory, a small wedge from the unit disc is also allotted. Then successive agents keep trimming the wedge so that the total piece allocated is less than *1/n* for each of the remaining agents. In order that allotted pieces do not end up having strips that break up other agents' allotments, a complicated procedure of secondary auctions involving rebidding of the same piece along with guidelines for trimming and growing the pieces make it difficult to implement any such scheme. Besides, the solution is still hobbled by the issues affecting Hill's proposition. In fact, there may be some extra implementation issues, like getting the appropriate functions for the conformal maps, which make it just as impractical as Hill's procedure. These functions depend on the shape of the land and hence have to be tailor made for each problem individually. Beck only proves the existence of such functions, and does not specify how they can be found.

Webb (May 1990) provides a combinatorial algorithm for the fair border problem based on Hill's existence results. The algorithm is recursive in nature and works as follows: A region R is bordered on all sides by n countries, $C_1, C_2, \ldots, C_n$. Each country has its own evaluation of the piece of land and draws a region R_i adjacent to itself such it is valued at *1/n* by it's own measure.

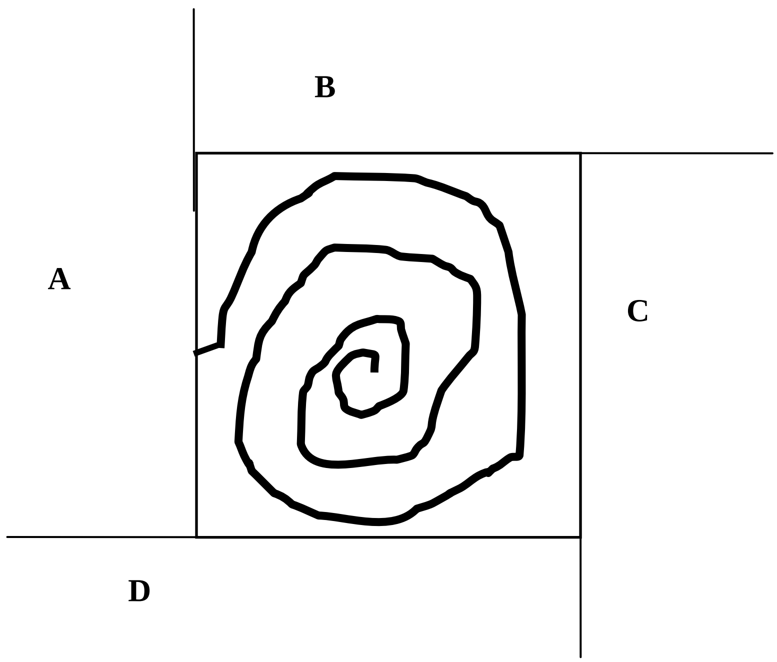


Figure 7: Agent A's demarcation of its share of the region. The thick line represents the land allocated to A

Then each country in turn (that values R_i greater than *1/n*) trims off a piece of R_i so as to disconnect it from C_i. The country that trims it last gets to keep the modified R_i. This region is attached to the allocated country by a strip of land small enough to be negligible to the others. The remaining land is then redistributed among the remaining agents similarly. The flexibility given by this algorithm in allowing agents to shape the region of interest as they like is also its drawback. If the shape of the land an agent gets is included in the fairness criterion then, earlier agents get a better deal than the later. This is because the later agents will have to "draw around" the regions allocated to earlier agents and the effective shape of the land they get might make it worthless for any use. Consider a square shaped region that is surrounded on four sides by four countries. Suppose agent A got the first chance to draw its region and it does so in the manner shown in Figure 7. It can be easily seen how the other agents' allotments are placed at a disadvantage.

Hill's paper provides the basis for another result (Maccheroni & Marinacci, 2003) that proves fair border solutions exist even if the utilities are concave. Utilities tend to be concave rather than additive in the real world. They take into account the fact that the marginal utility of a good decreases as consume more of it is consumed, due to satiation. Classic cake-cutting algorithms have always assumed that utilities were some sort of probability measure (Bartle, 1995). Consider a measurable space Σ as the resource being allocated. If A and B are two portions of the resource and v is the utility function of the agent, then we have the following property:

$$v(A \bigcup B) = v(A) + v(B) \quad \textit{for all disjoint } A,B \in \sum$$

On the other hand, a concave capacity has the property:

$$v(A \bigcup B) \leq v(A) + v(B) \quad \textit{for all disjoint } A,B \in \sum$$

Concave utility measures can be easily generated by wrapping a probability measure with a strictly increasing "utility" function u:

$$v(A) = u(\mu(A))$$

The authors were able to prove that the Dubins and Spanier solution holds in the concave domain too. Similarly they also showed that Hill's fair border solution is true for concave utilities. The authors do not however mention some benefits of the concave domain. Concave functions are useful because they connect efficiency and fairness. If the valuation function is additive, one of the efficient allocations is to give the whole cake to one agent. Thus efficiency need not induce fairness. If the valuation function is concave however, it is more efficient to give pieces of the cake to different agents since the whole cake allocated one agent is less valuable (in terms of social welfare) than the sum of the values that each agent attributes to the piece allocated to them. Thus an efficient algorithm will also tend to be fairer to all agents. The authors were however not able to determine if the same results would hold if the domain was subadditive rather than concave. (Concave domains are a subset of subadditive domains.) Finally the biggest drawback of the paper is that there is no constructive algorithm offered to actually apportion resources.

A novel approach (Thomson, Oct 2005) to resource allocation is to model the preference relations over the set of arcs on a circular cake geometrically. The paper considers the resource to be an infinitely divisible, non-homogeneous and atomless one-dimensional continuum whose end-points are topologically identified. It then proceeds to partition the resource into intervals. It is assumed that the recipients are equipped with preferences over intervals that are additive, continuous and monotonic with respect to interval inclusion. The assumptions about the domain as well as the preference relations are quite similar to the ones in our earlier paper. There are, however, a number of restrictions on the preferences:

- The preferences should be smooth, i.e., they should have continuously differentiable numerical representations. While this is a simple requirement, it is not trivial to have a preference that is "kink-free" due to the circular shape of the cake.
- Preferences should be convex. Again this is a non-trivial condition as convexity depends on the choice of origins.

The paper places emphasis on egalitarian-equivalence solutions rather than envy-free solutions as the former tend to have more pareto-efficient solutions than the latter. Egalitarian-equivalence is a distributional requirement that states that there exists a reference consumption that each agent finds indifferent to its own consumption. Egalitarian-equivalence is however a weaker condition than envy-freeness. Thus all envy-free solutions are egalitarian-equivalent while the converse is not true. Since in many domains the set of efficient solutions are not envy free, egalitarian-equivalence is used as a “compromise” fairness requirement. Other limitations in this paper are that the analysis of preference relations over the union of two or more intervals becomes quite complex. Trying to model preference relations over the union of at most two intervals requires four-dimensional space. This paper is limited to the domain of one-dimensional resource allocation and does not offer a constructive algorithm for allocating intervals. One of the salient points is that explicit utility functions are not required. Finally the paper also discusses strategy-proofness of their solutions. A rule is strategy-proof if no agent ever has an incentive to misrepresent its preferences. Unfortunately for the case of $n=2$ agents it turns out that any pareto optimal solution that is strategy-proof is also dictatorial i.e., any one particular agent will always be able to get its most preferred choice of interval irrespective of other agents’ choices.

Chambers (2005) discusses the various normative properties that land division rules should have based on the principle of utilitarianism. Utilitarianism(Mill & Sher, 2002) is a principle in the theory of ethics that prescribes the quantitative maximization of beneficial consequences for a population. The paper studies the circumstances in which rules can satisfy the division independence property. If the union of the portions allocated to an agent from various subparcels of a parcel is identical to the portion allocated from the initial parcel, then the rule based on which the allocation is made is said to be division independent. This property may be attractive in case a large piece of resource needs to be broken down into smaller pieces in order to make the allocation problem more tractable. Alternatively if the quantity of a resource keeps increasing over time and the allocation needs to be done over each additional portion incrementally, a rule that is division independent is desirable. The paper goes on to show that any rule that satisfies the above property along with a few other normative properties like independence of infeasible land and efficiency is a subrule of the weighted utilitarian rule. This is useful since utilitarian social welfare functions have been well studied in literature and can be used to get information about the system. However the division independence requirement is too constraining and invalidates a lot of other useful properties. For example, division independence is mutually exclusive with the “Strong positive treatment of equals” property. This is discouraging because the above property is a necessary condition for fair solutions to exist. This is a feature of weighted utilitarian rules in general and if the rules are required to be scale invariant as well (i.e., the portions allocated by the rule are independent of the scale of the utility function), then only dictatorial rules are possible. Thus ironically the social welfare approach ends up empowering one individual at the expense of the rest. The other issues are:

- Division independence is meaningful only in the context of additive functions. If the utilities are concave then utility of the union of subparcels will be less than the sum of the utilities of the individual subparcels and, hence, there is no notion of division independence.
- The weighted utilitarian rules require the cardinal comparison of utilities to work. This assumption is not tenable in real world situations.
- The definition of utilitarianism is too vague and cannot be fixed by an objective function.

It is thus clear that division independence is too tight a constraint to be of any significant use in evaluating competing allocation algorithms.

The literature survey shows the multifaceted nature of the land division problem. Researchers have focused on various aspects of the problem to get a better handle on how it can be resolved. However each has limitations; either due to the approach taken (measure-theoretic vs. combinatorial), or due to the nature of the domain. This shows that the problem is complex. Also as mentioned earlier, the land-division literature essentially assumes that the resource is a flat two-dimensional plane. Since we are tackling a two-dimensional cylindrical shaped resource, new approaches and techniques are called for. In the next section we begin a description of the details of our approach and how it stands up in comparison to ideas in the existing literature.

3. Dividing a Two-Dimensional Cylindrical Resource among *n* Agents

Our earlier work (Iyer & Huhns, Oct 2005) presented solutions for one-dimensional linear RA and one-dimensional circular RA problems. Based on how the protocols and procedures are set up, we can exploit those results for the cylindrical RA problem. The first section presents the framework for transforming the cylindrical RA to circular RA problem. The next section proposes a less restrictive protocol and procedures that make use of results in the linear RA and planar RA research to come up with an expanded space of solutions. The protocol is a simple one-shot type of protocol. At the end of the negotiation either the procedure is able to find a solution and every agent gets a fair share of the resource, or the procedure is unable to find a solution and a conflict deal is reached i.e., no agent gets any part of the resource. After the agents have indicated their preferences, a mediator will run the procedure to allocate the resources. Who is the mediator? The mediator can be chosen from among the participating agents themselves. Because our procedure is verifiable (i.e., any agent who wants to double check the allocation process can simply run the procedure again after obtaining preference information of other agents), a mediator does not need to possess special qualifications like being unbiased for example. Thus mediator bias does not become a factor because the allocation procedure is deterministic. As the mediator is chosen from among the agents, information of agent preferences is not revealed to external parties and better privacy is afforded.
A cylindrical resource has two distinct dimensions. One dimension is parallel to the axis of the cylinder and is finite and bounded. This is referred to as the Linear dimension (L-D). The other dimension is perpendicular to the axis of the cylinder and runs along the circumference of the circle that forms the cross-section of the cylinder. This dimension is finite and unbounded. This is referred to as the Circular dimension (C-D). Refer Figure 8.

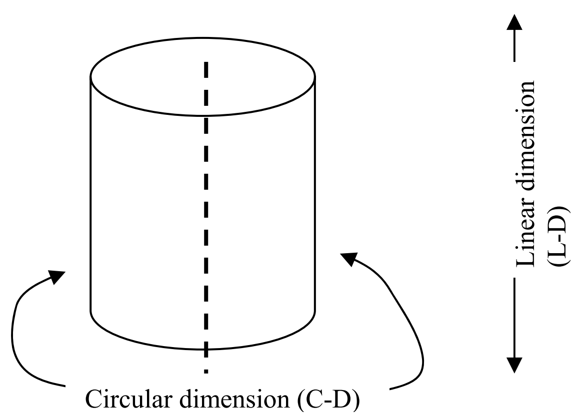


Figure 8: The dimensions of a cylindrical resource.

3.1 Mapping the Cylindrical RA Problem to the Circular RA Problem

The following protocol is a modification of the protocol specified for the circular RA problem by Iyer and Huhns (Oct 2005).

Circular RA Protocol

If there are *n* agents,
1. Each agent will create *n+1* portions of the resource all of which will be equal by its valuation.
2. Agents can create portions only by marking rectangles on the cylindrical surface. Other shapes not allowed.
3. Portions created by the same agent should not overlap with each other along the C-D.

Thus *n* agents will demarcate *n+1* non overlapping (along the C-D) rectangles creating a total of *n(n+1)* rectangles. If every agent follows the protocol then fair allotments are guaranteed for every agent.

Circular RA Procedure

1. Project the rectangles onto the C-D. This will look like closed intervals on the C-D. Refer Figure 9.
2. Choose a start mark on the C-D and move in the counterclockwise direction (both chosen arbitrarily).
3. Extend the end point of each interval till the start point of the next interval belonging to the same agent.
4. We have thus transformed this problem into a circular RA problem. The procedure for allocating resources was presented by Iyer and Huhns (Oct 2005). This procedure can be applied to obtain a fair allocation for all agents.

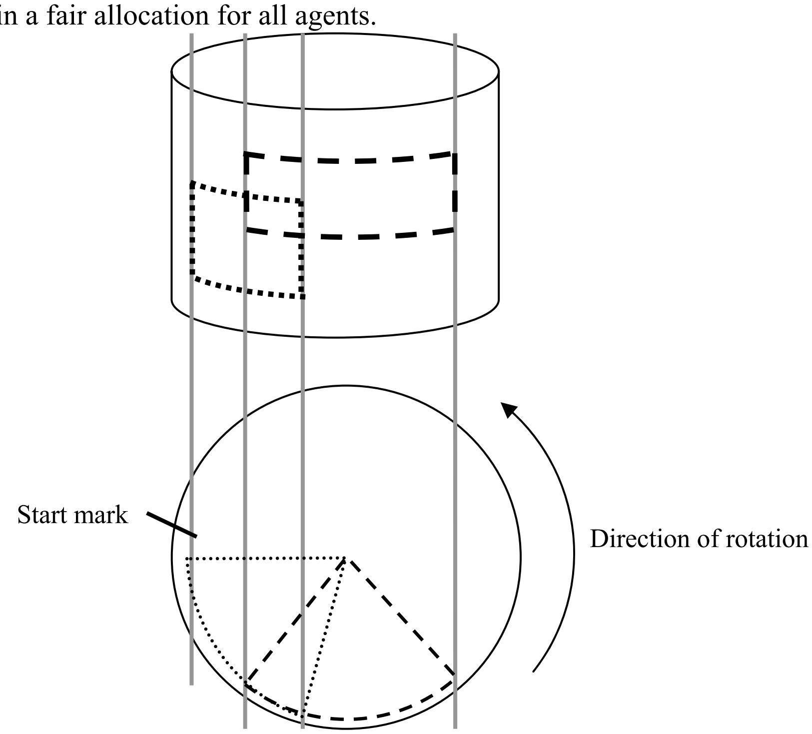


Figure 9: Projection of rectangles onto the circular dimension.

The following theorem guarantees that each agent will receive a fair share that of the resource.

Theorem 3.1. If there are *n* agents and each agent makes *n+1* radial marks, creating *n+1* portions of a circular resource, then the procedure guarantees that each agent will be allotted a portion of the resource, such that the portion is one of the *n+1* portions created by the agent itself.
The proof and the procedure for this theorem are described by Iyer and Huhns (Oct 2005).

Features

The procedure is fair in the sense that if agents adhere to the protocol, then each agent is guaranteed a portion of the resource which was demarcated by the agent itself. However since each agent creates *n+1* equal portions of the resource, the portion it receives will be worth at most *1/(n+1)* of the entire resource. This makes it inefficient in comparison to other procedures which strive to give each agent at least *1/n* of the entire resource. However the advantage of this procedure is that no mediator bias in involved as the procedure is completely deterministic and can be verified by any agent by running the procedure again.

3.2 Mapping the Cylindrical RA Problem to the Linear RA Problem

In this section, we present a protocol that enables the mapping of the cylindrical RA into the linear RA. We call this the *cylindrical RA protocol* because the same protocol can be for the native two-dimensional cylindrical RA procedure that will be presented later.

Cylindrical RA Protocol

If there are *n* agents,

1. Then each agent will create *n* portions of the resource, all of which will be equal by its valuation.
2. Agents can create portions only by marking rectangles on the cylindrical surface. Other shapes are not allowed.
3. Portions created by the same agent should not overlap with each other.

In order to guarantee an allocation, it has to be verified that agent preferences fulfill some qualifying conditions.

Condition A

For every agent, the rectangles marked out by a particular agent do not overlap along the L-D.

If condition A is true then, we can map the two-dimensional cylindrical resource allocation problem (cylindrical RA) into a one-dimensional linear resource allocation problem (linear RA). Procedures for solving a linear RA problem were proposed in our earlier paper (Iyer & Huhns, Oct 2005) and can be applied in this case. We present the procedure for allocating the resource by the following example. Consider two agents, labeled *i* and *j* each marking two rectangles of equal value (as per the requirement of the protocol). See Figure 10.

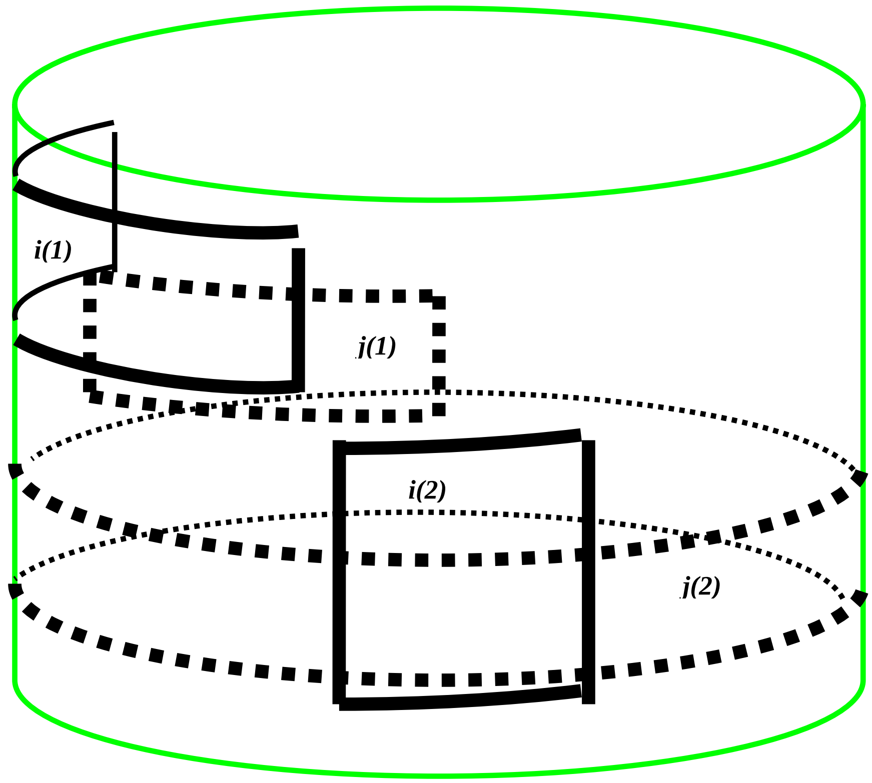


Figure 10: Agents *i* and *j* mark out two rectangles each on the cylindrical surface. Agent *i*'s rectangles are represented by solid lines, while agent *j*'s rectangles are marked by the dotted lines

Procedure given condition A

Due to condition A, it is known that rectangles belonging to a particular agent do not overlap along the L-axis. The following procedure can then be applied.

1. Project the rectangles along the L-axis. These will now look like closed intervals on this axis (see Figure 11).
2. Starting from the bottom and moving to the top, extend the start point of the interval of each agent to the end point of the previous interval (belonging to the same agent)
3. The last mark (when moving from bottom to top) of every agent is extended to the boundary of the resource (see Figure 12).
4. We have thus transformed the cylindrical RA problem into a linear RA problem. The procedure for allocating resources was presented by Iyer and Huhns (Oct 2005). This procedure can be applied to obtain a fair allocation for all agents. Proceeding from bottom to top (an arbitrary choice) and moving along the L-axis, it is now possible to guarantee each agent will get a rectangle that belongs to it.

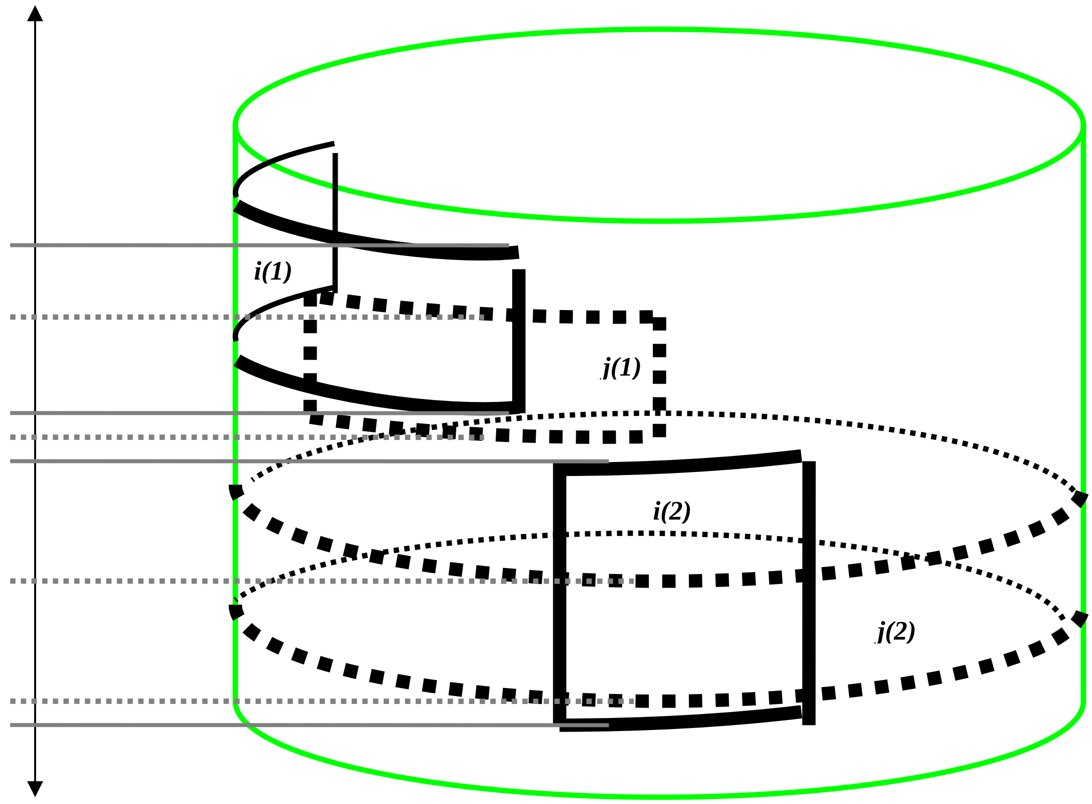


Figure 11: Projections of the agents' portions onto the L-axis

The following theorem guarantees that each agent will receive a fair share that of the resource.

Theorem 3.2. If there are *n* agents and each agent makes *n-1* marks, creating *n* portions of a linear resource, then the procedure guarantees that each agent will be allotted a portion of the resource, such that the portion is one of the *n* portions created by the agent itself.

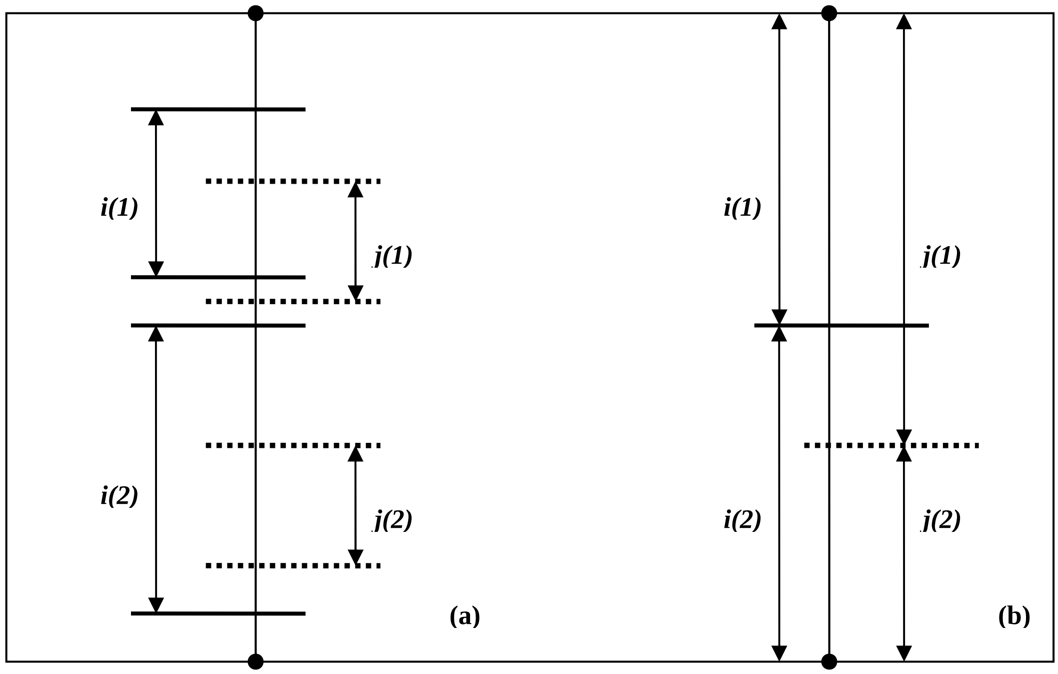


Figure 12: Transforming the two-dimensional cylindrical RA into the one-dimensional linear RA

The proof and the procedure for this theorem is described by Iyer and Huhns (Oct 2005).

Features of procedure for condition A

The procedure for condition A modifies agent intervals. This is done in order to map the cylindrical RA problem to the linear RA problem. However this modification is benign because the modified intervals are effectively the original intervals padded with zero utility regions. Once the allocation procedure terminates, we can easily obtain the original intervals by removing the padded regions (in case the padded region is of negative utility to the agent). We exploit the property that if the co-ordinates of any two rectangles made by a particular agent do not overlap along the L-axis, then the rectangles do not overlap at all (even if the co-ordinates overlap along the C-axis). By extending the intervals along the L-axis, we ensure that entire L-axis is exhaustively used up. Now the problem is equivalent to the linear RA problem which has been tackled in by Iyer and Huhns (Oct 2005).

3.3 A Native Cylindrical RA Procedure

The protocol required for the native cylindrical RA procedure has already been mentioned in the previous section. Condition A is restrictive on how agents should mark their rectangles. Condition B is put forth to allow agents greater flexibility in how agents should draw rectangles on the surface of the cylindrical resource while still guaranteeing an allocation can exist. If none of the above conditions hold true then our procedure (for condition B) can still be run to check for the presence of a feasible allocation but no guarantees can be made. The procedure is novel because it takes into account the topology of the overlaps of agent preferences on a cylindrical surface. An example showing resource allocation among three competing agents gives an intuition of the procedure. This is followed by proof that the procedure is guaranteed to find an allocation when condition B is true. We begin by stating condition B:

Condition B

Each rectangle must have a degree of partial overlap at most equal to one.

Condition B permits greater flexibility in how agents can mark their rectangles. As per the cylindrical RA protocol, it is still required that rectangles belonging to the same agent do not overlap. The sufficient condition for this is that the corresponding intervals along the L-axis or the C-axis do not overlap. Note that condition A was more rigid because the overlaps of intervals were to be avoided specifically on the L-axis. Agents specify their preferences (by marking out rectangles) to a mediator. The mediator then determines whether condition B is fulfilled. If so, then a solution can be guaranteed and the allocation procedure is executed.
The notion of the degree of partial overlap was proposed in by Iyer and Huhns (Feb 2007b). The notion of overlap is explained first. Figure 13 shows two rectangles drawn on the surface of the cylinder that overlap each other. The corresponding intervals are projected on to the L-axis and the C-axis. The following types of overlap (relationship), between any two intervals, can occur along each axis:

1. Separate(S): The intervals do not overlap.
2. Partial (P): There exists a partial overlap between the intervals.
3. Superset/Subset (Sp/Sb): One interval is completely enclosed by the other.

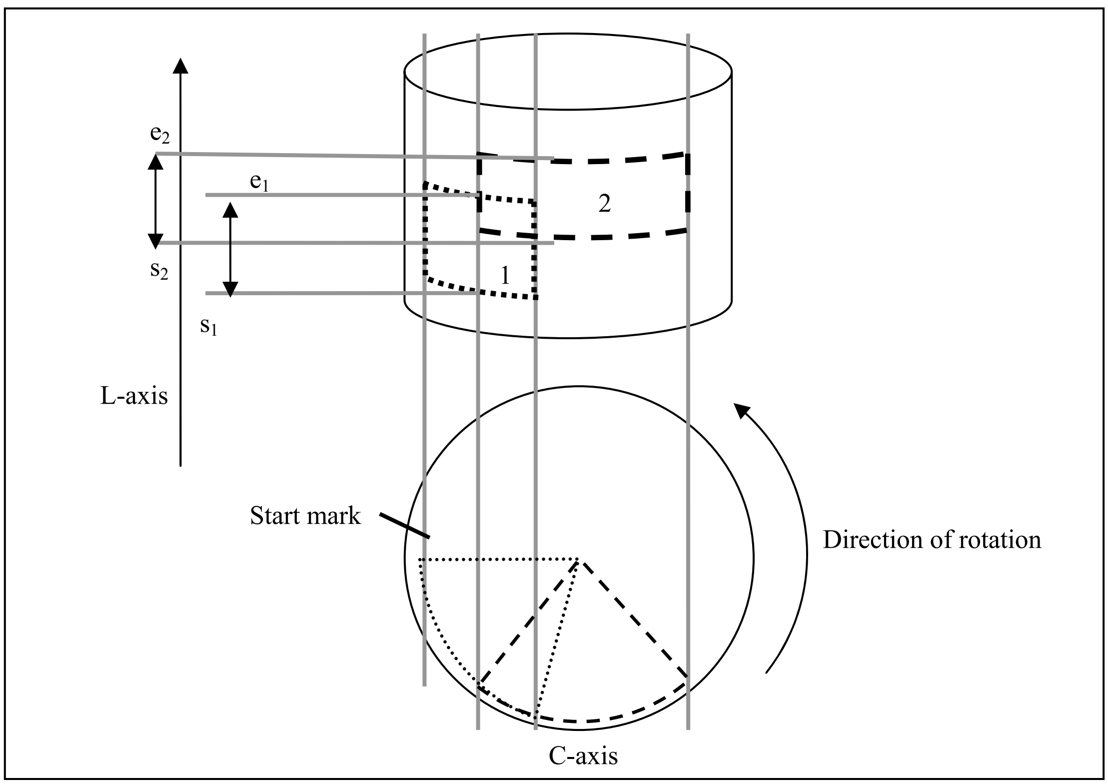


Figure 13: Nature of overlap of agent rectangles along the C-axis and the L-axis

The relationship between any two rectangles can be completely determined by the relationship between the corresponding intervals along the L-axis and C-axis respectively. The location of a rectangle is completely specified by two values along the L-axis (L-points) that form the L-interval and the two values along the C-axis (C-points) that form the C-interval. Hence once the procedure has a list of L-interval and C-interval positions, it should be able to determine the relationship between any two rectangles. Given any two pairs of L-interval points, it is possible to find the relationship between the corresponding L-intervals. We choose an arbitrary convention of moving from bottom to top along the L-axis, reading off L-points we go. The start point of an interval is denoted as s and the end is denoted as e. The subscript denotes the label of the rectangle that is specified by the points. Figure 14 shows why the s_1- s_2- e_1- e_2 combination of points is denoted as a Partial overlap between intervals. The table below shows the various permutations of points and the corresponding relationship between the intervals. Topologically only a few of the combination of points are unique. We exploit the following property to weed out redundancies:

Swap invariance property

The relationship between intervals is invariant with respect to the swapping of their respective start labels and end labels.

	Ordering	**Relationship**
1	s_1- s_2- e_1- e_2	P
2	s_1- s_2- e_2- e_1	Sp/Sb
3	s_1- e_1- s_2- e_2	S
4	s_2- s_1- e_1- e_2	Sp/Sb
5	s_2- s_1- e_2- e_1	P
6	s_2- e_2- s_1- e_1	S

Table 1: Relationships between two L-intervals based on the ordering of the points

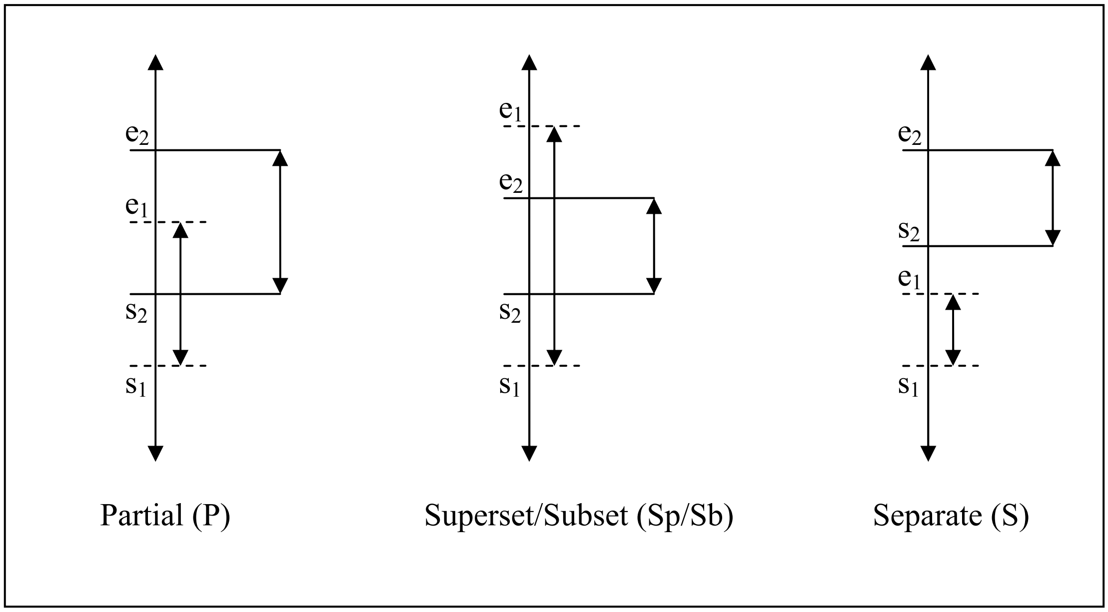


Figure 14: The topologically unique overlaps of L-intervals

Thus s_1 ←→ s_2 and e_1 ←→ e_2 does not affect the relationship between intervals. For example, if we swap start labels and end labels in case 1: s_1- s_2- e_1- e_2 becomes s_2- s_1- e_2- e_1 which is case 5 (a partial overlap). The swapping property holds for any of the above combination of points. This symmetry reduces the number of combinations of interest to 3 unique types. Cases 1, 2 and 3 are the only topologically unique combinations that exist. They map to the three type of overlaps P, Sp/Sb and S respectively and are shown in

The relationship of the intervals along the C-axis is bit more complicated. The following conventions are used:

- We move in the counterclockwise direction while recording marks.
- We mark an arbitrary position on the circumference as the start mark.

Note that in the case of a linear dimension (L-axis) the end point(e_1) of an interval would not be encountered before the start point(s_1). But because a closed loop (C-axis) is topologically distinct from a linear dimension, end points can occur before start points.

Figure 15 shows an example where e_1 is read in before s_1 and how the interval is to be interpreted. The 24 combinations of s_1, e_1, s_2, e_2 are described in Table 2 along with the relationships they imply.

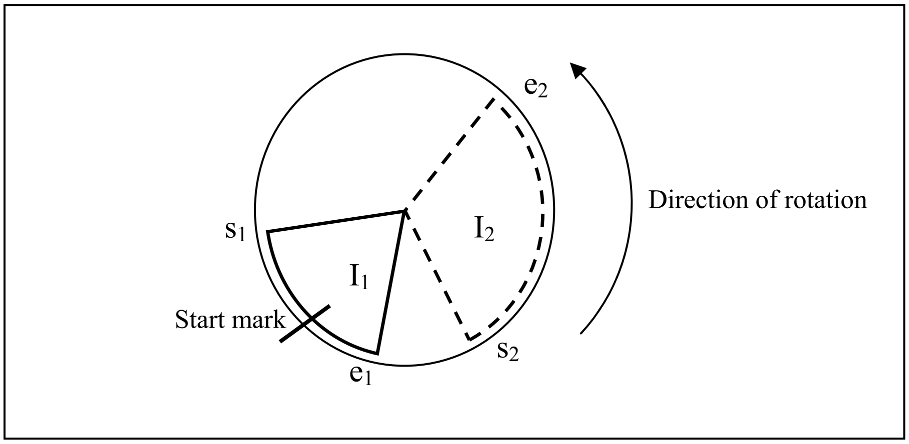
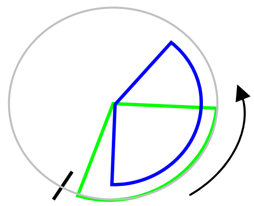
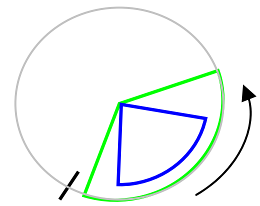
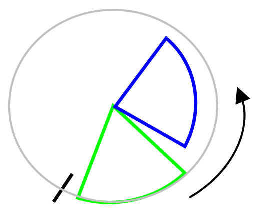
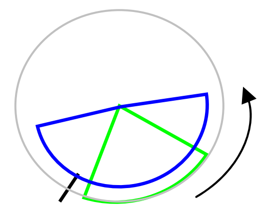
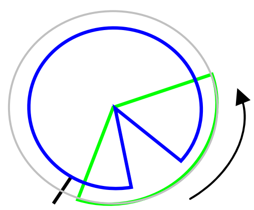
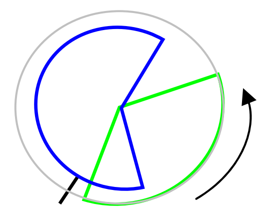
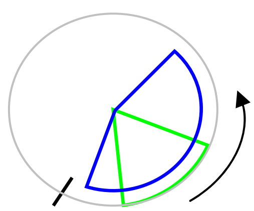
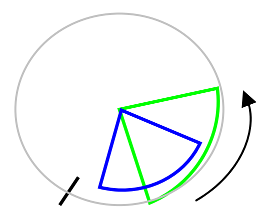
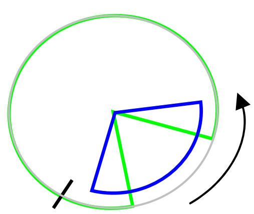
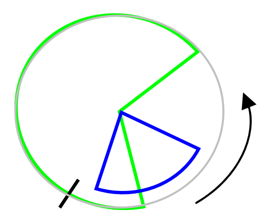
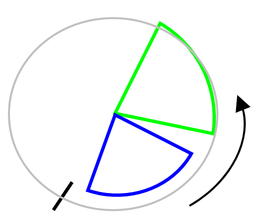
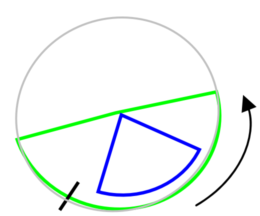


Figure 15: Interpreting an interval: Based on the placement of the start mark and direction of rotation, it is possible to encounter the end mark (e_1) of an interval before the start mark (s_1). The sequence of points maps to case 16 in Table 2 and the deduced intervals I_1 and I_2 are shown

Again in this case to we can apply the swap invariance property to reduce the number of topologically unique combinations to 12. Table 3 shows the unique combinations. The 12 combinations for the C-axis along with the 3 combinations for the L-axis yield 36 combinations of overlaps for rectangles on surface of the cylinder. Figure 16 displays 9 such combinations based on which case is picked for the L-axis and C-axis respectively.

Case No.	Ordering	Relationship		Case No.	Ordering	Relationship	
1	s_1-s_2-e_1-e_2		P	2	s_1-s_2-e_2-e_1		Sp/Sb
3	s_1-e_1-s_2-e_2		S	4	s_1-e_1-e_2-s_2		Sp/Sb
5	s_1-e_2-s_2-e_1		P	6	s_1-e_2-e_1-s_2		P
7	s_2-s_1-e_1-e_2		Sp/Sb	8	s_2-s_1-e_2-e_1		P
9	s_2-e_1-s_1-e_2		P	10	s_2-e_1-e_2-s_1		P
11	s_2-e_2-s_1-e_1		S	12	s_2-e_2-e_1-s_1		Sp/Sb

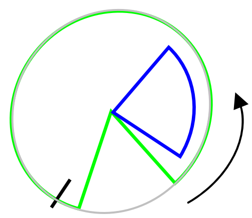
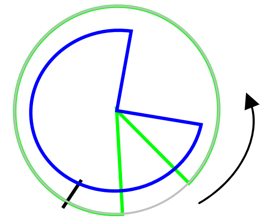
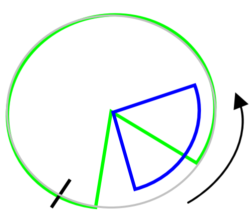
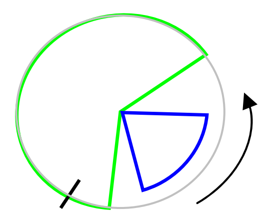
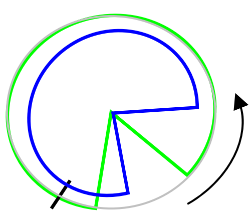
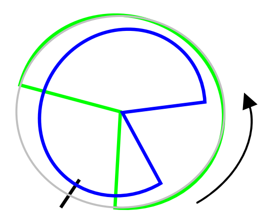
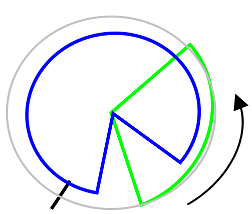
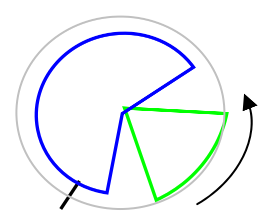
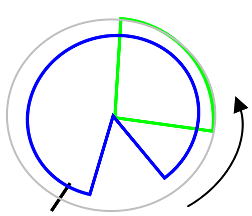
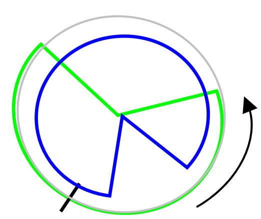
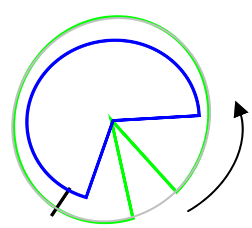
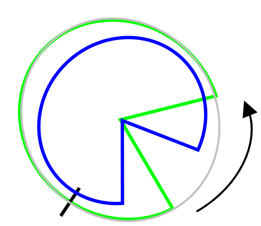
Case No.	Ordering	Relationship		Case No.	Ordering	Relationship	
13	e_1-s_1-s_2-e_2		Sp/Sb	14	e_1-s_1-e_2-s_2		P
15	e_1-s_2-s_1-e_2		P	16	e_1-s_2-e_2-s_1		S
17	e_1-e_2-s_1-s_2		P	18	e_1-e_2-s_2-s_1		P
19	e_2-s_1-s_2-e_1		P	20	e_2-s_1-e_1-s_2		S
21	e_2-s_2-s_1-e_1		P	22	e_2-s_2-e_1-s_1		P
23	e_2-e_1-s_1-s_2		Sp/Sb	24	e_2-e_1-s_2-s_1		P

Table 2: Relationships between two C-intervals based on the ordering of the points

Case No	Interval label			Case No	Interval label	
1	s_1-s_2-e_1-e_2	P		13	e_1-s_1-s_2-e_2	Sp/Sb
2	s_1-s_2-e_2-e_1	Sp/Sb		14	e_1-s_1-e_2-s_2	P
3	s_1-e_1-s_2-e_2	S		15	e_1-s_2-s_1-e_2	P
4	s_1-e_1-e_2-s_2	Sp/Sb		16	e_1-s_2-e_2-s_1	S
5	s_1-e_2-s_2-e_1	P		17	e_1-e_2-s_1-s_2	P
6	s_1-e_2-e_1-s_2	P		18	e_1-e_2-s_2-s_1	P

Table 3: The 12 topologically unique cases of overlaps along the C-axis

The following definitions set up the framework to explain the concept of degree of partial overlap.

*Definition 1***:** Any two rectangles that overlap with each other are *neighbors* of each other.

*Definition 2***:** Neighbors that overlap each other partially are *partial neighbors*.

Case 9 in Figure 16 is an example of two rectangles that are neighbors of each other but not partial neighbors, whereas cases 5, 6, and 8 show examples of partial neighbors.

*Definition 3***:** The *degree of partial overlap* of a rectangle is the largest number of partial neighbors it has, such that all these neighbors belong to the same agent.

By verifying that the degree of partial overlap is not more than one, it can be ensured that not more than one rectangle of each agent overlaps partially with the rectangle under consideration. If this condition is fulfilled, then there exists a procedure that is guaranteed to allocate a rectangle to each agent such that the rectangle was marked by the agent itself. This procedure was originally put forth by Iyer and Huhns (Feb 2007b) to solve the allocation of a two-dimensional planar resource. We illustrate the procedure through an example.

3.4 An Example Allocation Problem

We continue with the running example of three agents *i,j,* and *k,* which are contesting for a portion of the two-dimensional cylindrical resource. The agents have to mark out three rectangles, each, which are equal in their valuation as per the protocol. Based on the topology of rectangle overlaps (Figure 17), we compute that the degree of partial overlap is at most one, i.e., condition B is fulfilled. Therefore the procedure will be able to find a feasible allocation.

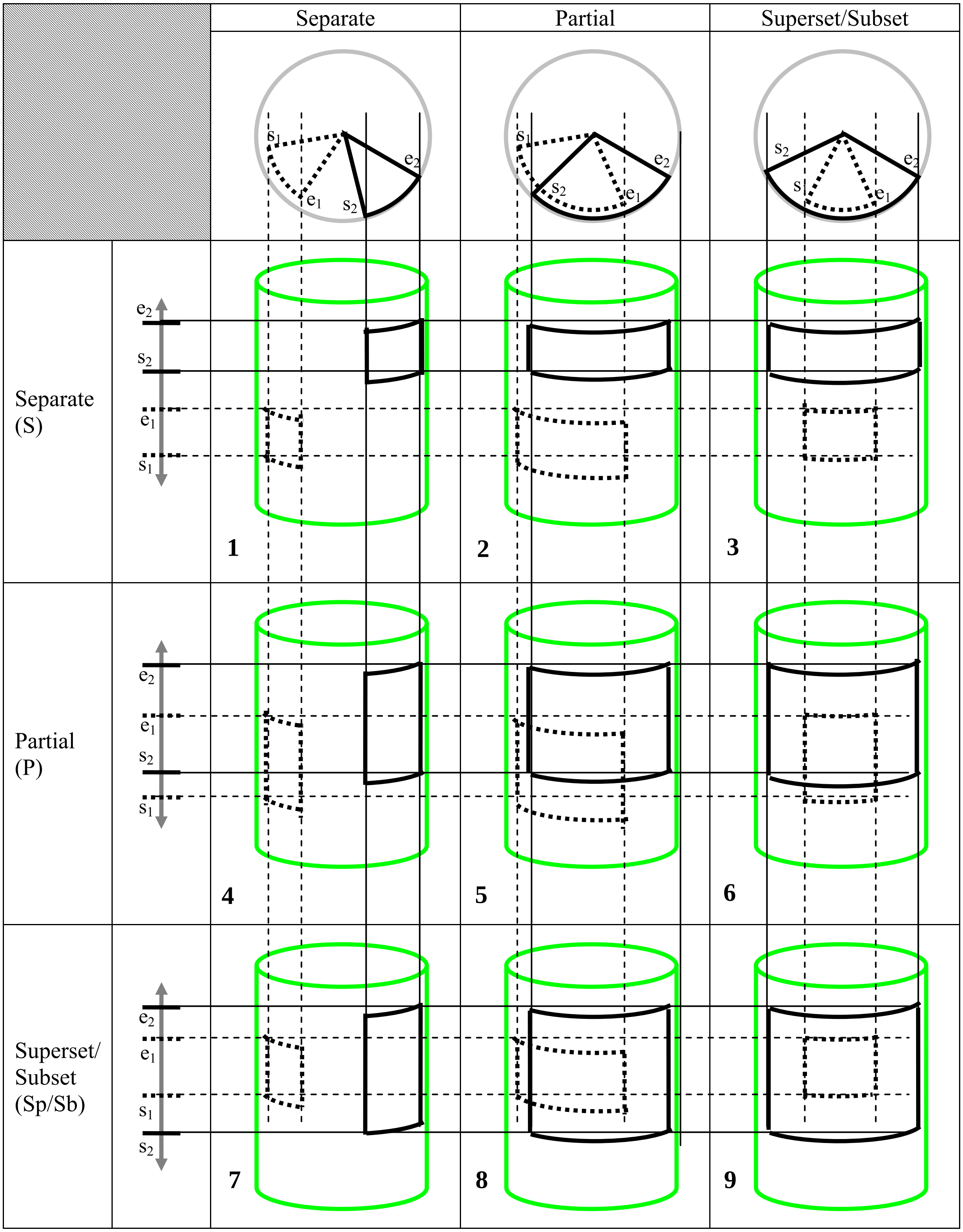


Figure 16: The different types of overlap between two rectangles on a cylindrical surface. Only 9 of the 36 possible combinations are shown.

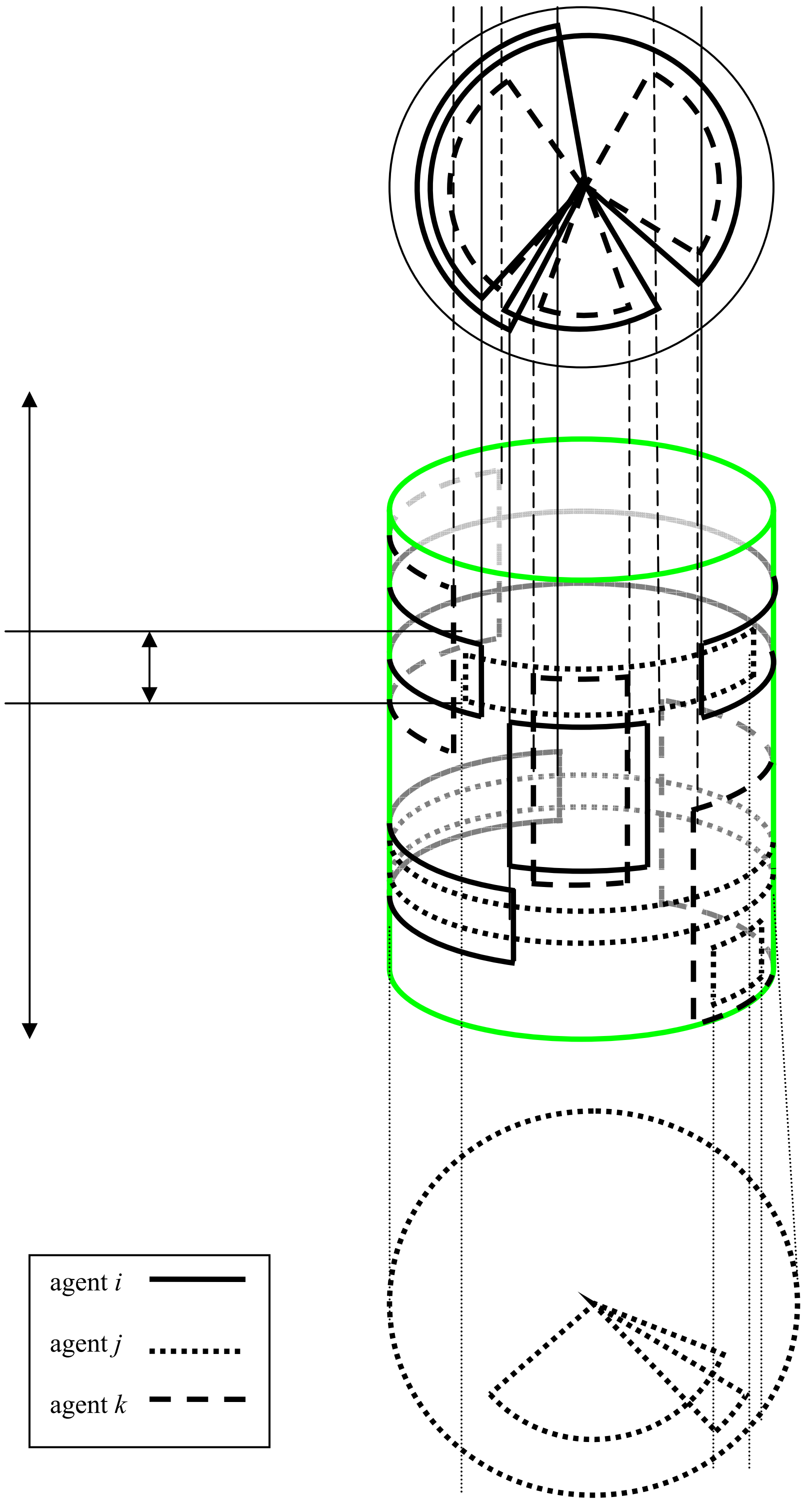


Figure 17: Agents ,*j,* and *k,* mark out portions on a cylindrical resource that fulfill condition B. All projections along the C-axis are shown. Only one projection onto the L-axis is shown to reduce clutter

The procedure given condition B will be executed as follows:

1. The rectangles are submitted to a mediator that collects them in a list.
2. The L-co-ordinates of the intervals are read into the L-list. The procedure sorts the L-co-ordinates of the intervals in the order of their occurrence from bottom to top. The C-co-ordinates are sorted by (arbitrarily) picking one C-co-ordinate as the first one. Then moving in the counterclockwise direction we read add the points that are encountered into the C-list. The procedure will determine which rectangle needs to be allotted to an agent based on the orderings of the intervals. The actual position of the points on the cylindrical surface is not needed by the allocation procedure. Since there are no cardinal comparisons between agent rectangles (i.e., "areas" of rectangles are not computed), we do not require the resource to be measurable (Bartle, 1995).
3. We determine the relation of the L-intervals to each other. The relations can be one of S, P or Sp/Sb. Thus each rectangle has a set of L-relations with its neighbors after we parse through the L-list. The procedure similarly processes C-list creating a set of C relations for each rectangle.
4. We utilize a scoring matrix as shown in Figure 18. Note that the matrix has 16 values although we mentioned 36 scenarios earlier. This is because, in order to determine the relationship between any two rectangles, we need only the relations they have with each other along the L-axis and C-axis respectively. And along each axis there exist only four possible values for the relations: S, P, Sp, Sb. Note that Sp(superset) and Sb(subset) are the same topology-wise. For example, if interval A is a superset (Sp) of interval B, then it is the same as saying interval B is the subset (Sb) of interval A. But the scoring matrix is created from the "point of view" of a particular rectangle. Thus rectangle A being a subset of rectangle B is distinct from rectangle B being a superset of rectangle C. Hence Sp and Sb are treated as separate cases. Also note that the sufficient condition for any two rectangles not to overlap is that their respective intervals do not overlap in at least one of the axes. Since such rectangles are not neighbors of each other they are not assigned a score.

	S	**P**	**Sp**	**Sb**
S	**--**	**--**	**--**	**--**
P	**--**	**0**	**0**	**0**
Sp	**--**	**0**	**-1**	**0**
Sb	**--**	**0**	**0**	**1**

Figure 18: The scoring matrix for a rectangle. The types of interval overlaps are Separate(S), Partial(P), Superset(Sp), Subset(Sb)

5. Based on the types of overlap each rectangle has with its neighbors, we create a directed graph that represents the connections. The directed graph is created as follows:

For any two rectangles,

   a. Each rectangle is represented as a node.

b. If there is no overlap between the two rectangles, there is no edge between the corresponding nodes.
c. If there is a partial overlap between the two rectangles, then each node will have a directed edge connecting the other node with an edge weight of 0.
d. If rectangle A is a subset of rectangle B, the edge going from Node B to Node A will have a weight of 1, whereas the edge going from Node A to Node B will have a weight of -1. Refer Figure 19 for the graph corresponding to the example in Figure 17.

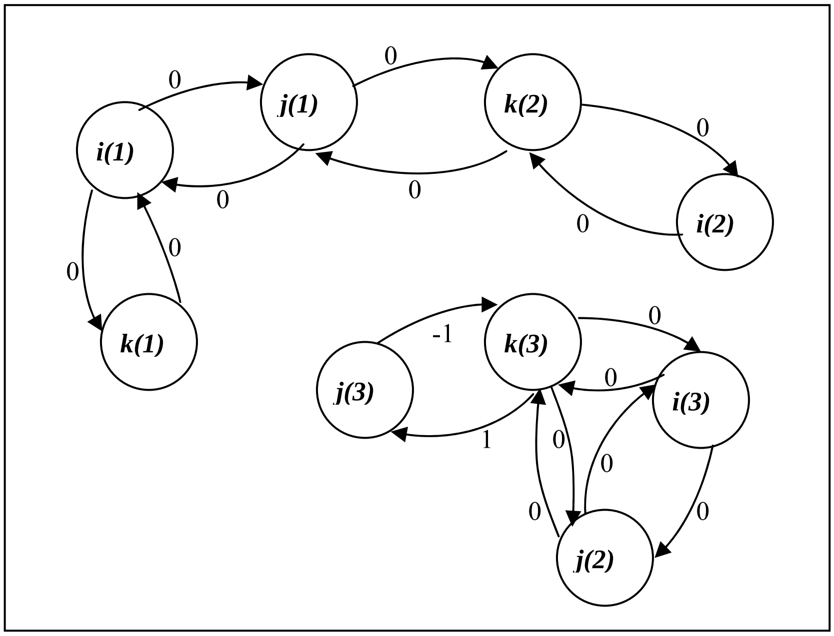


Figure 19: The directed graph based on the topology of overlapping rectangles

6. Once we have the topology of the connections in graph form, the procedure will arbitrarily start at some rectangle (i.e., node) and allot the particular rectangle based on the following conditions:
   a. If there is no outgoing edge with edge weight of 1, then allocate this particular node.
   b. If there is such an edge, then travel along this edge and move to the node connected to it. Now let this be the new node under consideration. Repeat the procedure using this node as the starting point.

Once a particular node (i.e., the rectangle corresponding to it) has been allocated then we remove the following nodes from the graph, for future consideration:

   c. The node just allocated.
   d. All neighbors of the current node.
   e. All the nodes belonging to the agent which is the owner of the current (allocated) node.

Repeat Step 6 till all agents have been allocated nodes (rectangles).

Applying step 6 to our example, we choose *i(1)* (arbitrarily) to be the first node. *i(1)* has no outgoing edge with edge weight 1, hence it is allocated (step 6a). Next we remove the following nodes:

- *i(1)* (Step 6c)
- *k(1)* and *j(1)* (Step 6d)
- *i(2), i(3)* (Step 6e)

The following nodes remain: *j(2),j(3),k(2),k(3)* and agents *j* and *k* still have to be allocated their nodes (rectangles). So we repeat Step 6 and arbitrarily choose *k(3)* as our starting node. Now *k(3)* has an outgoing edge with edge weight 1, so we travel along the edge and move to node *j(3)*. *j(3)* has no outgoing edges with weight 1, so allocate this node. Next applying Steps 6c, 6d, and 6e we remove the following nodes from consideration: *j(3), k(3), j(2)*. Only node *k(3)* is left which can be allocated to agent *k*. Thus each agent receives a fair share of the resource.
The proof that the procedure is guaranteed to come up with a fair allocation is described by Iyer and Huhns (Feb 2007b). Only the statement of the theorem is mentioned here.

Theorem 3.3. If there are *n* agents and each agent makes *n* rectangles, creating *n* portions of a cake and if the rectangles are so marked that the degree of partial overlap is not greater than 1, then our procedure guarantees that each agent will be allotted a piece, such that the piece was one of the *n* portions created by the agent itself.

The following pseudo code gives an idea about the structure of the procedure:

```
while(graph.hasMoreNodes())
   currentNode=graph.getNewNode()
   while(currentNode.hasMoreEdges())
     currentEdge=currentNode.getNewEdge()
       if(currentEdge.getEdgeWeight()==1)
         currentNode=currentEdge.getOtherNode(currentNode)
       end if
   end while hasMoreEdges
   currenNode.setAllocated(true)
   graph.remove(currentNode)
   graph .remove(currentNode.getNeighbors())
   graph.remove(currentNode.getOwner.getOwnedRectangles())
end while hasMoreNodes
```

No two nodes belonging to the same agent will ever share a common edge in the graph. This is because the agents have to adhere to the protocol and ensure that no two rectangles belonging to the same agent overlap. The procedure will be executed by the agent which volunteers to be the mediator. The agent will keep a record of the list of nodes it has visited. This allows for any other agent to run the procedure and confirm that the list of nodes traveled is valid. This is how one can verify that the allocation is fair and guard against mediator bias. Note that the set of allocated nodes depends on which node is chosen as the starting node. But it does not affect the feasibility of the allocation. Even though we choose the starting node arbitrarily, we will always be guaranteed a feasible allocation because it was confirmed that the degree of partial overlap was at most 1.
We conclude this section with some brief discussion on the features of this procedure. This procedure is similar to the hill-climbing algorithm (Russell & Norvig) that tries to find the

"highest point" among various nodes. The procedure is also less restrictive in terms of how agents mark their rectangles. Once the mediator has verified that the degree of overlap is at most 1, a solution can be guaranteed. In case this condition is false, one can still run the procedure, but there is no guarantee that a feasible solution exists. The degree of partial overlap property is a only a sufficient condition for the existence of a solution, it is not a necessary condition.

4. Discussion

A detailed analysis of the various issues involved with the allocation of a two-dimensional planar resource is available by Iyer and Huhns (Feb 2007b). Most of that discussion is applicable to the two-dimensional cylindrical resource. We present a summary of the earlier discussion for context. One issue is to study how solution space looks like with respect to condition A and condition B. Figure 20 shows the relationship between solution spaces for conditions A and B. This can be interpreted as:

- Condition A and condition B are not subsets of each other.
- Condition A and condition B are not mutually exclusive.

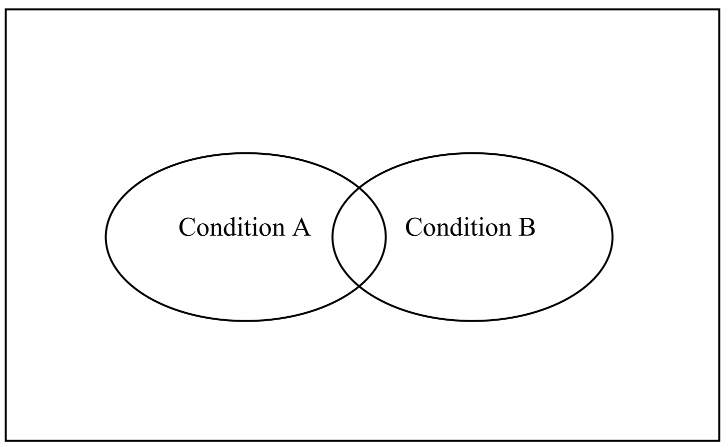


Figure 20: The solution space for conditions A and B

Thus condition B is more flexible, but is not a weaker condition. It however helps in expanding the space of solutions available to us. Another issue of interest is the existence of solutions in case conditions A and B are false. Is it possible that if the agents follow just the protocol, that some algorithm can guarantee a feasible allocation? The following theorem answers in the negative:

Theorem 4.1. If the agents follow only the protocol, then there exists no algorithm that can guarantee an allocation of rectangles, such that each agent will get a rectangle marked by itself.

Iyer and Huhns (Feb 2007b) provide a description of the proof. This is unlike the results for the one-dimensional linear RA problem and and circular RA problem (Iyer & Huhns, Oct 2005) where a feasible allocation is guaranteed simply by ensuring that agents follow the protocol. One can, however, devise iterative versions of the cylindrical RA protocol, which begin with allocation of rectangles (even though they overlap with each other) and then apportion progressively smaller regions of dispute (i.e., the parts where the rectangles overlap), in every iteration. But complications can arise in case of some resources like land where incontiguous pieces are of little value to any agent.

Next, we discuss how well the procedure compares with various criteria mentioned in the literature survey. Specific references are cited for each criterion in case the reader needs further clarification of the concepts.

- **Fairness** (Brams & Taylor, 1996)**:** The proposed protocol is fair. As mentioned earlier in the paper, an agent will find the allocation *fair,* if it feels it got exactly *1/n* of the value of the entire resource. Since as per the protocol the agent divides the resource into *n* rectangles of equal value and the procedure (given condition A and/or B is true) will allot one such rectangle to that agent, the agent gets $1/n^{th}$ (by its own valuation) of the entire resource. Thus our procedure allocates resources in a fair manner.
- **Envy-freeness** (Brams & Taylor, 1996)**:** Envy-freeness means that every agent thinks that the portion allocated to it is greater in value than the portions other agents received. Thus no agent envies another agent's portion. It is strongly desirable for procedures to be envy-free but it is tougher criteria to fulfill than just fairness. There exist very few procedures that can guarantee envy-free division. Our procedure while fair is not envy-free. It is quite possible that though an agent may consider the value of the portion it received to be *1/n* of the total (and hence fair), it may be envious of another agent whose portion it thinks is of greater value than its own.
- **Efficiency:** Efficiency is also a criterion for judging procedures. It provides a measure of how much of the resource is wasted in the process of allocation. We use the term "efficient" in the pareto optimal sense, i.e., is it possible that there exists an alternative allocation in which at least one agent is better off while the others are no worse than the current allocation? The answer in the case of our procedure is yes, i.e., our procedure is inefficient at allocating resources. This is because all the portions allocated to agents do not exhaustively use up the resource. The "leftovers" can be redistributed amongst agents by running the protocol again and letting interested agents submit their preferences. This can be done repeatedly until finally every agent thinks the remnants are negligible in value. Such an iterative version of the protocol will certainly help improve the efficiency of resource distribution. However the effectiveness of such an approach depends on the type of resource being distributed. If the resource is *infinitely divisible* and *recombinable* (like a cake) then such an iterative procedure is appealing. But some resources like scheduling time on a supercomputer users or land division may only be infinitely divisible (though not recombinable) and in such cases the repeated divisions of the leftovers will make the agents portions so small as to be basically worthless. Our protocol and procedure is however more efficient than the traditional one-dimensional cake-cutting, especially when agents value some portions of the resource negatively. This is because we allow the agents to cut in two dimensions, thus completely eliminating negative utility regions from being included in the allocation. While there is a clear need to improve the efficiency of our single shot protocol and procedure, it needs to be matched up against the concurrent increase in the complexity of the protocol as well as the procedure.
- **Complexity:** What is the space and time complexity of this procedure? Let us calculate the space complexity first. Each rectangle is fixed by four points, with each point having two values (one each for the C-coordinates and L-coordinates respectively). Thus each rectangle is described by eight values. Each agent marks *n* rectangles, creating a total of n^2 rectangles (by *n* agents). Thus the space complexity of the procedure is $8n^2$**.** Now we compute the time complexity. First the list of C-coordinates and L-coordinates needs to be sorted. An efficient sorting algorithm, such as insertion sort, will take *nlog n* time, which translates to $2n^2 log\ n$ time (replacing *n* by n^2) for each axis. Next C and L relations need to be created. Since

condition B is true, each rectangle can have at most *n-1* partial neighbors. If one rectangle is completely subsumes the other (i.e., they are neighbors but not partial neighbors), then the inner rectangle can have at most *n-2* partial neighbors. Thus the rectangles will have an average of *n-1* neighbors in the worst case. Thus a total of $n(n-1)= n^2-n$ relations are possible on each axis. Analogously the graph created out of these relations will have vertices equal to n^2 and number of edges equal to n^2-n (in the worst case). The running time of the procedure to traverse these nodes is therefore $O(n^2)$.

- **Strategy-proof** (Rosenschein & Zlotkin, 1994)**:** A protocol is said to be *strategy-proof,* if for each agent declaring its true evaluation is a dominant strategy. A typical example of a strategy-proof mechanism is the Vickery auction where agents submit sealed bids to the auctioneer and the agent quoting the highest price wins but is only required to pay the amount of the second highest bid. Our protocol is strategy-proof. The dominant strategy for any agent while drawing out rectangles is to make sure that it values each of the rectangles it draws as exactly *1/n.* If an agent tries to draw a rectangle, that is smaller in value than any other agents' rectangle, in order to guarantee itself a specific piece of the resource, it may end up getting the piece, but the value of that piece will be smaller than *1/n,* thus getting less than its fair share. On the other hand, if an agent tries to get more than *1/n* by drawing a rectangle as large as possible, then it is quite likely that it will subset some other agents' rectangle which will end up getting allocated to another agent. For clarification consider the following extreme example: Agent *i* is greedy and wants as much of the resource as possible. So it draws one large rectangle which covers the entire resource, while other agents follow the dominant strategy and draw *n* rectangles of equal value. What will agent *i* receive? Agent *i* receives nothing because the procedure that allots rectangles takes as input one large rectangle (which agent *i* had drawn) and *n-1* rectangles of zero value and the procedure ends up allocating one as these zero valued rectangles as agent *i'*s allocation. Thus agent *i* gets zero utility by being greedy. Our protocol is therefore not open to manipulation by greedy agents and attempts to misreport preferences will backfire. There is no incentive for the agents to lie and hence the protocol is strategy-proof.
- **Measurability** (Bartle, 1995)**:** This is an important notion for mathematicians and economists. The basic assumption is that any resource being allocated must be "measurable", i.e., there must exist a function that can assign a number (like "length", "area" or "volume") to subsets of a given set (like land for example). This notion paves the way for a cardinal comparison of various subsets for enabling an allocation. However our procedure does not need cardinal comparisons to make an allocation. We create allocate portions based on the topology of overlaps. If the one rectangle is completely contained in another then the smaller rectangle is allocated. If two rectangles only partially overlap each other then the one that was encountered first (by the procedure) is allocated first. We do away with the notion of comparison of agent rectangles based on their "areas". While agents may need the set to be measurable in order to "measure" out equal portions of the resource, such a restriction is not imposed by the protocol itself. Agents may use alternate means to create equal portions, like creating *n* rectangles arbitrarily and adjusting their sizes iteratively till it doesn't prefer any one over the other. This is a way to *ordinally* create *n* rectangles of equal value without using the notion of a "measure". Unlike earlier work in land division, we do not explicitly require that the resource to be allocated be measurable.
- **Constructive (Non-existential):** The procedure proposed here is algorithmic. Most of the early literature dealing with two-dimensional resources (primarily in the form of land division) is dominated by economists and mathematicians. Their results are primarily existential in nature viz. they analyze various features of the problem (like types of utility functions, fairness and efficiency of the possible allocations etc) and show whether or not

allocations "exist" with given properties. They do not however propose ways to find such solutions. Thus it is yet unknown that even if feasible solutions exist whether the problem of finding an allocation procedure is tractable. Because of a lack of constructive solutions to allocation problems it is not possible to implement them in the real world where agents negotiate with one another and a mediator is able to compute a feasible allocation in a reasonable amount of time. For multiagent system designers, in addition to the knowledge that feasible allocations exist, the following points need to be considered:

(a) Is there a procedure to find these allocations? (Yes)
(b) What is the communication cost of the procedure? ($O(n^2)$)
(c) What is the computational complexity of the procedure? ($O(n^2)$)
(d) Can the agent preferences be kept private? (Yes)
(e) Can the procedure be manipulated by agents' lies? (No)
(f) What is the efficiency of the allocation procedure? (Inefficient)
(g) Are the procedure and the protocol iterative? (No)
(h) Are the agents restricted in using certain classes of utility functions? (No)

We answer some of the questions in this paper. Ours is the first result that can be expressed as a computer algorithm for allocation of two-dimensional cylindrical resources. Not only does it test for the existence of a solution, it is able to find one if it exists (given condition A and/or B is true).

- **Nature of agent utility functions:** The survey of earlier literature shows a number of restrictions placed on agent utility functions none of which are applicable in our case. The agents may draw rectangles based on their internal utility functions or completely do away with the use of utility functions. As stated in the discussion earlier about measurability, agents can also draw rectangles using ordinal rather than cardinal comparisons. Briefly, we mention the typical restrictions on agent utility functions historically mentioned in literature, which are *not* applicable to our case:

(a) Non-atomic: This requirement specifies that the agent utility functions smoothly go down to zero as the amount of the resource tends to zero, i.e., there should be no "atoms" in the resource, where portions smaller than the "atom" are of zero value to the agent.

(b) Additive: This is common requirement for utility functions and helps simplify a lot of analysis. If A and B are two disjoint subsets then simple additivity states that:

$$v(A \bigcup B) = v(A) + v(B) \quad \textit{for all disjoint } A,B \in \sum$$

(c) Concave: Concave domains are a subset of subadditive domains and the utlity functions have the property:

$$v(A \bigcup B) \leq v(A) + v(B) \quad \textit{for all disjoint } A,B \in \sum$$

They are less commonly modeled than simple addtivity because it is difficult to prove results in this domain. However they more realistically reflect the utility of obtaining additional resources in the real world.

Continuous: Utility functions are assumed to be continuous. This means that for every amount x of the resource the agent has a unique value for that amount. This simplifies analysis but may not be necessarily true in the real world. In addition, utility functions are said to be smooth if there are no "kinks" in the function value anywhere i.e., the function is differentiable at all points of the curve.

5. Future Work

This paper presents protocols and procedures for the allocation of a two-dimensional cylindrical resource. Our earlier work proposed an allocation scheme for one-dimensional linear RA and circular RA problem (Iyer & Huhns, Oct 2005) as well as the two-dimensional planar RA problem (Iyer & Huhns, Feb 2007b). There is however a lot of scope for future works with this paper as a starting point. The protocols and procedures can be extended to apply to other domains, such as resources that are topologically different from a cylinder (a torus, or a sphere for example). Further research will also help improve the procedure by reducing the running times and allowing more generalized shapes. We mention the pertinent issues that need to be looked at:

- **Increasing the space of feasible allocations:** The protocol in its current form is a single shot protocol. Either the agents rectangles are laid out in a manner where feasible allocations exist or no agent gets any part of the resource (a conflict deal is reached). There is a need for creating an iterative version of the protocol whereby agents can send proposals to one another by continuously changing the shape of their rectangles and finding a solution that is feasible (and preferably optimal) to all the agents. It is quite likely that there will be a concomitant increase in the complexity of the protocol, communications, and the procedure. Thus while the space of solutions will certainly increase, mechanism designers will have to balance this against increased cost of computation both for the agents as well as the mediator.
- **Search for condition X:** A more generic condition than the two we have proposed will serve to increase the space of feasible allocations. Conditions A and B serve as a benchmark against which future conditions can be compared. Conditions A and B are *sufficient* but not *necessary* conditions for the guaranteeing the existence of a solution. There can exist other conditions which behave similarly. The ideal condition X would be one that is both sufficient and necessary. In such a case condition X would exhaustively cover the solution space and would be the (largest) superset of both solutions covered by conditions A and B.
- ***n*-dimensional resources:** In this paper we considered the case where a resource is a generalized two-dimensional cylinder. We extended our earlier result of one-dimensional resource allocation to two dimensions. But there were significant qualitative differences between the problems. We presented proof that if condition B is true then a feasible allocation exists and the procedure would be able to find the allocation. One can fashion proofs in an analogous manner for three (and higher) dimensional resources. But the challenge is to prove that the proof for condition B holds for arbitrary *n*-dimensional resources. Is it possible to create a (meta) proof to prove a feasible allocation exists — if condition B is true — for an *n*-dimensional resource with arbitrary *n*?
- **Efficiency:** The current protocol while being more efficient than one dimensional resource allocation schemes (like moving knife) is inefficient because there will always be leftovers from the allocation. As mentioned earlier, efficiency can be improved by creating a simple iterative version of the single shot protocol. Any work done to improve the efficiency of allocation will greatly increase the appeal of this procedure but it is likely to increase the complexity of the protocol as well the procedure in the process.
- **Allowing more general shapes:** The protocol as it is presented currently allows agent to mark regions of interest in the shape of rectangles only. This no doubt simplifies the allocation procedure but it is quite restrictive. If agents were allowed to mark out regions of interest as more general polygons then it will improve the efficiency of the system and increase the utility received by agents. It may also increase the space of feasible allocations. This is a challenging problem. If a procedure were found which is able to allocate for general

polygons then it might be possible to extrapolate those results to the case where agents can draw amoeba shaped regions and the procedure can find a satisfactory solution.

- **Allowing other topologies:** We assume the resource to be a two-dimensional cylindrical surface. Would it be possible to extrapolate the procedure to other shapes like the surfaces of spheres or torii? Our procedures are not directly applicable to such shapes because of their different topology. But one can easily visualize resources having such topologies; for example: dividing underwater mineral resources may require the agents to take into account that the earth has the topology of a sphere rather than approximate it as a flat plane.
- **Creating a decentralized version of the procedure:** The procedure in its current form is centralized. Since the running time is $O(n^2)$, it is quite appealing to create a distributed form of this procedure, so that running time of the procedure is reduced. While creating a distributed version of the procedure is a non-trivial task in itself, the following issues need to be considered as well:

a) Is a different set of agents needed to execute the distributed procedure? Or can the participating agents themselves execute the procedure?
b) How can the information about agent preferences be kept private?

6. Conclusion

This paper presents protocols and procedures for allocation of two-dimensional cylindrical resources. These are resources that have the topology of the cylindrical surface. Such resources are encountered when one of their dimensions is finite and bounded (the axial dimension) while the other is finite and unbounded (the circular dimension). The paper presented some real world examples to show the practical nature of the problem. The kind of cylindrical surfaces that were amenable to our protocol and procedure was discussed. Existing literature does yet explicitly cover such cases. It was shown how a cylindrical resource allocation problem could be mapped into the one-dimensional circular resource allocation problem and the appropriate procedure from our earlier work could be applied. Next, a more flexible protocol was proposed wherein agents only marked n non-overlapping rectangles of equal value (by their valuation) on the cylindrical surface of the resource. If condition A was true, then we could transform the existing allocation problem to a one-dimensional linear resource allocation problem which has already been solved. If condition B is true we present a novel two-dimensional cylindrical resource allocation procedure which is guaranteed to come up with a solution. It uses the notion of degree of partial overlap to create a sufficiency condition for the existence of a solution, and proposes a procedure to come up with one in such a case. The discussion shows that the solution spaces for condition A and condition B overlap with each other partially. Thus each condition serves to expand the space of available solutions. The proposed solution is fair, strategy-proof, constructive and does not need the resource to be measurable. There exists considerable scope for future work, both in terms of improving the performance of the proposed procedure as well extending this work to apply to other topologies and domains.

References

Austin, A. K. (Oct 1982). Sharing a cake. *Mathematical Gazette, 66*, 212-215.

Bartle, R. G. (1995). *The Elements of Integration and Lebesgue Measure*. New York: Wiley-Interscience.

Beck, A. (Feb 1987). Constructing a Fair Border. *American Mathematical Monthly, 94*, 157-162.

Brams, S. J., & Taylor, A. D. (1996). *Fair Division: From Cake-Cutting to Dispute Resolution*. New York: Cambridge University Press.

Bredin, J., Maheswaran, R. T., Imer, Ç., Basar, T., Kotz, D., & Rus, D. (June 2000). *A Game-Theoretic Formulation of Multi-Agent Resource Allocation.* Paper presented at the Fourth International Conference on Autonomous Agents.

Chambers, C. P. (2005). Allocation rules for land division. *Journal of Economic Theory, 121*, 236-258.

Dubins, L. E., & Spanier, E. H. (Jan 1961). How to cut a cake fairly. *American Mathematical Monthly, 68*, 1-17.

Goeree, J. K., Holt, C. A., & Ledyard, J. O. (July 2006). Report On Results Of Economic Experiments Examining Performance Properties Of Simultaneous Multiple Round Spectrum License Auctions With And Without Combinatorial Bidding Tech. Rep. Federal Communications Commission.

He, M., & Leung, H.-f. (2002). Agents in E-Commerce: State of the Art. *Knowledge and Information Systems, 4*(3), 257-282.

Henle, M. (1994). *A Combinatorial Introduction to Topology*. Mineola, NY: Dover Publications.

Hill, T. P. (Oct 1982). Sharing a cake. *Mathematical Gazette, 66*, 212-215.

Huhns, M. N., & Malhotra, A. K. (July 1999). Negotiating for Goods and Services. *IEEE Internet Computing, 3*(4), 97-99.

Iyer, K., & Huhns, M. (Oct 2005). *Multiagent Negotiation for Fair and Unbiased Resource Allocation.* Paper presented at the OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2005.

Iyer, K., & Huhns, M. N. (Feb 2007a). Negotiation Criteria for Multiagent Resource Allocation. Tech. Rep. TR-2007-004, Department of Computer Science and Engineering, University of South Carolina.

Iyer, K., & Huhns, M. N. (Feb 2007b). A Procedure for the Allocation of Two-Dimensional Resources in a Multiagent System. Tech. Rep. TR-2007-003, Department of Computer Science and Engineering, University of South Carolina.

Maccheroni, F., & Marinacci, M. (2003). How to cut a pizza fairly: Fair division with decreasing marginal evaluations. *Social Choice and Welfare, 20*(3), 457-465.

Mill, J. S., & Sher, G. (2002). *Utilitarianism*: Hackett Publishing Company.

Robertson, J., & Webb, W. (1998). *Cake Cutting Algorithms*. Nattick, MA: A K Peters Ltd.

Rosenschein, J. S., & Zlotkin, G. (1994). *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. London, England: MIT Press.

Russell, S. J., & Norvig, P. (2002). *Artificial Intelligence: A Modern Approach* (2nd edition ed.). NJ: Prentice Hall.

Sen, S., & Biswas, A. (2000). *More than Envy-Free.* Paper presented at the ICMAS'00.

Soh, L.-K., & Tsatsoulis, C. (Nov 2005). A Real-Time Negotiation Model and A Multi-Agent Sensor Network Implementation. *Autonomous Agents and Multi-Agent Systems, 11*(3), 215-271.

Steinhaus, H. (1948). The problem of fair division. *Econometrica, 16*, 101-104.

Stewart, I. (December 1998). Your Half 's Bigger Than My Half!. *Scientific American,* 112-114.

Tasnádi, A. (2003). A new proportional procedure for the n-person cake-cutting problem. *Economics Bulletin, 4*(33), 1-3.

Thomson, W. (Oct 2005). Children crying at birthday parties. Why? Fairness and incentives for cake division problems Tech. Rep. 526,

Webb, W. A. (May 1990). A Combinatorial Algorithm to Establish a Fair Border. *European Journal of Combinatorics, 11*, 301-304.