

MPRA

Munich Personal RePEc Archive

Non rivalry and complementarity in computer software

Marengo, Luigi and Pasquali, Corrado
Dime

2006

Online at <http://mpa.ub.uni-muenchen.de/25589/>
MPRA Paper No. 25589, posted 01. October 2010 / 16:08



Dynamics of Institutions and Markets in Europe is a network of excellence of social scientists in Europe, working on the economic and social consequences of increasing globalization and the rise of the knowledge economy.
<http://www.dime-eu.org/>

DIME Working Papers on INTELLECTUAL PROPERTY RIGHTS

http://ipr.dime-eu.org/ipr_publications

Emerging out of DIME Working Pack:

'The Rules, Norms and Standards on Knowledge Exchange'

Further information on the DIME IPR research and activities:

<http://ipr.dime-eu.org/>

This working paper is submitted by:

Luigi Marengo

SSSUP, Pisa, Italy.
Email: Luigi.Marengo@sssup.it

Corrado Pasquali

Università di Teramo, Italy.
Email: cpasquali@unite.it

Non Rivalry and Complementarity in Computer Software

**This is Working Paper
No 11 (April 2006)**

The Intellectual Property Rights (IPR) elements of the DIME Network currently focus on research in the area of patents, copyrights and related rights. DIME's IPR research is at the forefront as it addresses and debates current political and controversial IPR issues that affect businesses, nations and societies today. These issues challenge state of the art thinking and the existing analytical frameworks that dominate theoretical IPR literature in the fields of economics, management, politics, law and regulation- theory.



Sponsored by the
6th Framework Programme
of the European Union

Non Rivalry and Complementarity in Computer Software

LUIGI MARENGO

SSSUP Pisa, Italy
luigi.marengo@sssup.it

CORRADO PASQUALI

Università di Teramo, Italy
cpasquali@unite.it

April 15, 2006

Abstract

In this paper we contend that – contrary to what argued by a vast part of the literature – computer software and, more in general, digital goods (i.e. symbolic strings on an electronic medium with some economic value) do not present the characteristics of a public good as they do not suffer from lack of rivalry and excludability any more than other durable goods which are regularly allocated on competitive markets.

We argue instead that the “market allocation problem” – if any – with digital goods does not arise from their public nature but from some peculiar characteristics of the production technology. The latter presents the nature of a typical problem solving activity as far as the production of the first unit is concerned, this means that innovative activities in computer software are characterized by high degrees of interdependencies, cumulateness, sequentiality, path dependence and, more in general, sub-optimality arising from imperfect problem decompositions. As far as the production of further units is concerned, we observe instead high (but not infinite) expansibility and perfect codification (lack of any tacit dimension) which make diffusion costs rapidly fall.

Given such claims, we argue that a standard “Coasian” approach to property rights, designed to cope with the externalities of semi-public goods may not be appropriate for computer software, as it may decrease both ex-ante incentives to innovation and ex-post efficiency of diffusion. On the other hand the institutional definition of property rights may strongly influence the patterns of technological evolution and division of labor in directions which are not necessarily optimal.

1 Introduction

Computer software is an archetypical example of a digital good (Quah 2003), that is a good which is made of information encoded into strings of characters (e.g. – in addition to computer programs – files encoding music, pictures or movies, but also genetic codes), and is therefore usually considered as a public or quasi-public good because of the non-rivalry of the information it embodies. A good is non-rival in consumption if the same unit of that good can be consumed by potentially infinite consumers without diminishing the consumption level of any of them.

In the case of non-rival goods competitive markets cannot be relied upon to yield price signals that lead to socially efficient outcomes with respect to production and distribution. The marginal cost of information is zero but if information is distributed at zero cost, as required in an efficient market, producers will not have an adequate incentive to produce it.

The “property solution” to the information production problem amounts to creating, assigning and enforcing private intellectual property rights. This has as its most immediate consequence the creation of an artificial scarcity which assures the appropriability of returns from investments (mainly in R&D) necessary to produce the first unit. A monopoly right to the commercial exploitation of an idea is offered in return for its disclosure. This institutional device allows the organization of market exchanges of transferable exploitation rights which, by assigning value to commercially exploitable ideas, create economic incentives for people and firms to creating new ones.

That “property” can be or actually is applied both as an incentive mechanism or as a device to assure the appropriability of returns is however questioned by a number of empirical and theoretical studies. On the other hand, the recent and increasingly straightforward adoption of “property” in the domain of information and knowledge has raised a number of doubts as to its effectiveness as a means to promote their creation and diffusion.

According to this perspective, in our paper we will try and examine two main issues related to IP protection when meant to be an application of the property rights paradigm to the domain of digital goods in general and computer software in particular.

First we will question the assumption that digital goods are really non-rival and argue that the degree of non-rivalry of digital goods stands quite far from the one of pure public goods and close – on the contrary – to the one we can find in many private goods. The peculiarity of digital goods does not reside so much in their non-rivalry but rather in the low cost of production of the units after the first, i.e. in its quasi “infinite expansibility” David (1992).

Second, if the “problem” is in the production technology and not in the good itself, we should examine more closely the conditions of production. In

this paper we suggest that the production of computer software can be usefully framed in terms of problem solving. Problem solving activities present some important features, namely non-monotonic interdependencies and cumulative-ness. In this paper we sketch a model of problem-solving technology which allows us to point out the potential inefficiency of property rights fragmentation.

2 (Non-)rivalry of digital goods

A pure public good, e.g. public security, is non-rival because my consumption of the good is compatible with the joint consumption of *the same* good of many (virtually infinitely many) other consumers. Note however that rivalry is a matter of degree: only goods which get destroyed with consumption are fully rival. I can share with other people the services provided by my car, my HiFi equipment and my TV set up to a given capacity. Interestingly enough the legislator does not seem to worry about inhibiting such manifestations of non-rivalry as car pooling or inviting friends at one's place to watch a soccer match on TV. Moreover the "up to a given capacity" clause applies also to pure public goods: non-rivalry does not imply zero marginal costs for all quantity intervals.

Note that in pure public goods a high degree of non-rivalry is matched by non-misurability, and actually this seems the crucial factor for market failure. It is very hard if not impossible to define and measure a unit of consumption for these goods and therefore it is very hard or impossible to make consumers pay proportionally to the quantity consumed. We could for instance define "one hour without being victim of a crime" as the standard unit of consumption of public security but information, measuring and, in general, transaction costs would be enormous. Moreover my enjoyment of some units of "one hour without being victim of a crime" is only loosely related to the production of public security, because public goods are typically "environmental" goods (very high cross-externalities with various economic and non-economic activities): people living in Aosta enjoy, on average, more "hours without being victim of a crime" than people living in some outskirts of Naples, though the effort exerted by the producer of public security is considerably higher in the latter location.

Now, digital goods are neither non-measurable (at least to an extent greater than the one of all durable goods for which we pay a general and unspecified right to consume their services) nor environmental (though they might in some cases have considerable network externalities) and seem therefore to be completely missing some very important defining features of public goods.

What is really peculiar of digital goods is that the marginal costs of duplication are very low (but non zero) and the technology for duplication is

accessible to nearly everybody, while the costs of producing the first unit are considerable.

3 Problem-solving, interdependencies and the dangers of rights' fragmentation

As well documented in a wide array of empirical studies, in the past two decades the domain of (technological) knowledge has been so finely divided by property claims – on essentially complementary pieces of information – that the cost of reassembling constituent rights in order to engage in further research seems to be charging a heavy burden on technological advance. In the realm of scientific and technological research, this has taken the form of a spiral of overlapping patent claims in the hands of different owners reaching ever further upstream.

In our opinion, the attitude towards fragmentation is much in line with a Coasian effort to create as many rights as there are markets. Actually, the co-extensiveness of markets and property rights is hailed by economic wisdom as the setting in which competition can promote efficiency at its best: no transaction costs coupled with well defined and perfectly exchangeable property rights lead to perfect allocative efficiency.

Ideally, in a perfectly Coasian world a market would exist for every right with an economic value (Coase 1960). This presupposes individual property rights to be perfectly (and costlessly) defined, perfectly (and costlessly) enforced and perfectly (and costlessly) exchangeable. In this way every inefficient allocation would be avoided. In turn, the whole argument presupposes the very possibility of a limitless separability of property rights and of an ever finer definition thereof.

National authorities, both in the US and the EU, have adopted an attitude towards patenting that clearly reflects these principles. As a matter of fact, IP rights are being granted on increasingly fragmented “chunks” of knowledge such as single genes, databases, algorithms or parts thereof.

As suggested by Coase himself, the finest possible property rights structure is very likely to induce less rather than more competition as underlying markets will be so thin, with respect to the number of agents involved, as to induce monopolistic behaviors and considerable transaction costs. It thus seems that, in the end, allocative inefficiencies might arise which are not less serious than those which a strong and fine-grained rights structure was meant to eliminate.

A fast growing literature, both in the economic and legal discipline, is currently debating and questioning the idea that “more property rights imply more efficiency” and the idea that “commons, however defined and practiced are tragic”. Under this respect, we will suggest that the problem-solving na-

ture of the innovative process in computer software implies that more property rights do not necessarily imply more innovation and that moreover the institutional arrangements of rights do not only impinge upon the speed of innovation but also upon its direction. More finely defined intellectual property rights may cause sub-optimal technological trajectories. More in general we submit that the efficiency and sustainability of resources management systems crucially depends on technological characteristics of resources themselves and on contractual and institutional patterns of their usage in precise historical moments.

There can be little contention that the production of new software presents the typical characteristics of problem-solving activities. Notably, it is a process of designing viable solutions in a huge combinatorial problem space, characterized by very diffused interdependencies.

As pointed out by (Simon 1969), problem-solving by boundedly rational agents must necessarily proceed by decomposing any large, complex and intractable problem into smaller sub-problems that can be solved independently, by promoting what could be called the division of problem-solving labor. At the same time, note that the extent of the division of problem-solving labor is limited by the existence of interdependencies. If sub-problem decomposition separates interdependent elements, then solving each sub-problem independently does not allow overall optimization.

As a consequence, in the presence of strong interdependencies, one cannot optimize a system by separately optimizing each element it is made of. Consider a problem that is made up of N elements and whose optimal solution is $x_1^*x_2^*\dots x_N^*$ while the current state is $x_1x_2\dots x_N$. In the presence of strong interdependencies, it might well be the case that some or even all the solutions of the $x_1x_2\dots x_i^*\dots x_N$ kind show a worse performance than the current one.¹

It is important to remark that the introduction of any decentralized interaction mechanism, such as a competitive market for each component does not solve the problem. For instance, if we assume that in our previous example each component x_i is traded in a competitive market, superior components x_i^* will never be selected. Thus, interdependencies undermine the effectiveness of the selection process as a device for adaptive optimization and they introduce forms of path-dependency with lock-in into sub-optimal states that do not originate from the frictions and costs connected to the selection mechanism, but from the internal complexity of the entities undergoing selection.

As (Simon 1969) pointed out, an optimal decomposition (i.e. a decomposition that divides into separate sub-problems all and only the elements that are independent from each other) can only be designed by someone who has

¹Note that this notion of interdependency differs from the notion of complementarity as sub-modularity as in (Milgrom & Roberts 1990). Here, in fact, we allow for the possibility that positive variations in one component can decrease the system's performance value.

a perfect knowledge of the problem (including its optimal solution). On the contrary, boundedly rational agents will normally be forced to design near-decompositions, that is decompositions that try to put together, within the same sub-problem, only those components whose interdependencies are (or, we shall add, agents believe to be) more important for the overall system performance. However, near-decompositions involve a fundamental trade-off: on the one hand, finer decompositions exploit the advantages of decentralized local adaptation, that is, the use of a selection mechanism for achieving coordination “for free” together with parallelism and adaptation speed. However, on the other hand, finer decompositions imply a higher probability that interdependent components are separated into different sub-problems and therefore cannot, in general, be optimally adjusted together. In this paper we provide a precise measure of this trade-off and show that, in the presence of widespread interdependencies, finer than optimal decompositions have an evolutionary advantage (in terms of adaptation speed), although they inevitably involve lock-in into sub-optimal solutions.

One way of expressing the limits that interdependencies pose to the division of problem-solving labor is that global performance signals are not able to effectively drive decentralized search in the problem space. Local moves in the “right direction” might well decrease the overall performance if some other elements are not properly tuned. As Simon puts it, since an entity (e.g. an organism in biology or an organization in economics) only receives feedback from the environment concerning the fitness of the whole entity, only under conditions of near independence can the usual selection processes work successfully for complex systems (Simon 2002, p. 593).

A further aspect concerns the property that, in general, the search space of a problem is not given exogenously, but is constructed by individuals and organizations as a subjective representation of the problem itself and through the very process of problem-solving which defines a focal framework for future representations. If the division of problem-solving labor is limited by interdependencies, the structure of interdependencies itself depends on how the problem is framed by problem-solvers. Sometimes problem-solvers make major leaps forward by reframing the same problem in a novel way. As shown by many case studies, major innovations often appear when various elements that were well-known are recombined and put together under a different perspective. Indeed, one can go as far as to say that it is the representation of a problem that determines its purported difficulty² and that one of the fundamental functions of organizations is precisely to implement collective representations of the problems they face.

²(Simon 1969) argues that “Solving a problem simply means representing it so as to make the solution transparent”.

In the following we present a formal model, drawn from (Marengo, Pasquali & Valente 2005) and (Marengo & Dosi 2005). The key issue and difficulty addressed by the model is the opacity of single entities' functional relations and the partial understanding of their context-dependent individual contributions in forming a solution to the problem at hand. The model accounts for the relationships between problem complexity, task decentralization and “problem solving” efficiency. The main findings show that in domains of highly interdependent entities, such as complementary patents, there are delicate trade-offs between the exploitation of the advantages of decentralization and the need to control for complex interdependencies and that optimal dynamic search paths usually are not generated by highly fragmented structures. At the same time a precise set of tools are introduced to compare the efficiency properties of different institutional arrangements with respect to their different degrees of fragmentation and enforceability of property rights.

4 The model's general structure

The idea behind this model is to have a very general set of minimal elements with which we can easily build and analyze, mainly computationally but partly also analytically, collective problem solving under different institutional regimes. The model is strictly *agent-based*, in the sense that the elementary building block is a problem-solving agent and all aggregate properties are the result of the interaction among heterogeneous agents. The rules of interaction define the institutional structure, among which property rules play a central role.

The model has three components: environment, agents and institutional structure.

The environment is the problem space, where some entities (objects, solutions, artifacts, etc.) are evaluated against an exogenously given value (“fitness”) function. The environment is largely unknown to agents and as we will see it plays only the role of providing a payoff feed-back for the learning process.

Individual problem-solving agents have an imperfect knowledge of the environment and are heterogeneous, in the sense that they may differ from each other in terms of problem knowledge, problem-solving ability and objectives (evaluation of different solutions).

Agents adaptively search for better solutions in a problem environment they do not know, but search is directed by their cognitive representation of the environment.

In particular, agents are characterized by the following elements:

1. an encoding of the possible solutions and their relevant features in some

(binary) alphabet. Such encoding can be complete (one-to-one) or, most commonly, incomplete (many-to-one) meaning that different objects are encoded as equivalent (typically because some relevant features are neglected) and therefore cannot be distinguished;

2. a decomposition of the problem-space, i.e. a conjecture of the interdependencies among features, which act as a template for the generation of new tentative solutions;
3. a subjective evaluation function, which may coincide with or differ from the “true” value function of the environment and/or may coincide with or differ from the evaluation function of other agents.

Given such three elements, an agent searches for better solutions in an adaptive fashion: given the current solution it looks for better one by modifying some of the blocks defined by its decomposition. The new solution is accepted if it increases the value of the agent’s evaluation function.

In general this search process ends in a local optimum of the evaluation function, as common in NK type models (Kauffman 1993). The number, positions and values of such local optima depend on all the above mentioned elements: encoding, decomposition and evaluation.

We can precisely compute such local optima, with their basins of attraction and the expected time (number of steps) taken to reach one of them from any initial condition. Therefore we can also easily construct indexes of the agent’s ability in any given problem environment.

Finally, by institutional structure we mean the set of “rules of the game” through which agents interact. In particular we will concentrate on property rules and on two aspects thereof:

1. is property defined upon the results of individual search for solutions or can solutions be freely used and improved upon? In the former case property can take the form of veto and/or royalty fees on solutions which are somehow similar to a proprietary one.
2. at which level of granularity are rights defined? over entire solutions or on separate modules thereof? and how “large” are these modules?

In what follows we spell out the details of the model and present some very preliminary results. A fully fledged exploration of a full range of institutional structures is under way.

5 The basic component: a problem-solving agent

We assume that solving a given problem requires the coordination of N atomic “elements” or “actions” or “pieces of knowledge”, which we generically call

components, each of which can assume some (finite) number of alternative states. For the sake of simplicity, in what follows we assume that each component can assume only two alternative states, labelled 0 and 1. Note that all the properties presented below for the two-states case can very easily be extended to any finite number of states.

More precisely, we characterize a problem by the following elements:

A finite set of **objects**, or possible solutions: $O = \{o_1, o_2, \dots, o_w\}$

An **ordering** over the set of objects: we write $o^i \succeq o^j$ (or $o^i \succ o^j$) whenever o^i is weakly (or strictly) preferred to o^j .

A **problem** is defined by the pair (O, \succeq) .

Problem-solving agents do not have a direct knowledge of the problem itself, but hold and use a representation thereof. Such a representation consists of an encoding of the objects and a decomposition of the encoded problem space.

An **encoding** of the set of objects is a mapping from the set O to a set of internal states: $\Omega : O \mapsto X$ where X is the set of words of an alphabet which we assume binary for simplicity: $C = \{c_1, c_2, \dots, c_N\}$ with $c_i \in \{0, 1\}$. All in all, $X = \{x^1, x^2, \dots, x^{2^N}\}$ is the set of encoded objects.

A representation is complete if the mapping Ω is one-to-one and therefore O has the same cardinality as X , is incomplete if it is many-to-one. In most relevant situations the sheer cardinality of the problem's search space makes complete representations unattainable by human beings. Individual problem-solvers are bound to use heuristics which vastly reduce the size of the search space: for instance in a Rubik cube players look only at one side of the cube, treating as identical all states for which that side's configuration is the same. Thus, while maintaining that assuming complete representations is unrealistic, we will anyway begin our analysis from there, in order to obtain some benchmark results against which we can then compare those derived from the more realistic assumption of incomplete representations.

Even when holding complete representations, and therefore conceiving the entire search space, the combinatorial nature of this space makes it much too vast to be extensively searched by agents with bounded computational capabilities. One way of reducing its size is to decompose³ it into sub-spaces.

Let $I = \{1, 2, \dots, N\}$ be the set of indexes and let a **block**⁴ $d_i \subseteq I$ be a non-empty subset of it, we call the **size of block** d_i , its cardinality $|d_i|$. We define a **decomposition scheme** (or simply **decomposition**) of the problem

³A decomposition can be considered as a particular case of search heuristics: search heuristics are, in fact, ways of reducing the number of configurations to be considered in a search process.

⁴Blocks in our model can be considered as a formalization of the notion of modules used by the flourishing literature on modularity in technologies and organizations (Baldwin & Clark 2000) and decomposition schemes are a formalization of the notion of system architecture which defines the set of modules in which a technological system or an organization are decomposed.

(X, \succeq) a set of blocks:

$$D = \{d_1, d_2, \dots, d_k\}$$

such that $\bigcup_{i=1}^k d_i = I$

Note that a decomposition does not necessarily have to be a partition.

Given a configuration x^i and a block d_j , we call block-configuration $x^i(d_j)$ the substring of length $|d_j|$ containing the components of configuration x^i belonging to block d_j :

$$x^i(d_j) = x_{j_1}^i x_{j_2}^i \dots x_{j_{|d_j|}}^i$$

for all $j_h \in d_j$.

We also use the notation $x^i(d_{-j})$ to indicate the substring of length $N - |d_j|$ containing the components of configuration x^i not belonging to block d_j .

Two block-configurations can be united into a larger block-configuration by means of the \vee operator so defined:

$$x(d_j) \vee y(d_h) = z(d_j \cup d_h) \text{ where } z_\nu = \begin{cases} x_\nu & \text{if } \nu \in d_j \\ y_\nu & \text{otherwise} \end{cases}$$

We can therefore write $x^i = x^i(d_j) \vee x^i(d_{-j})$ for any d_j .

Moreover, we define the **size of a decomposition scheme** as the size of its largest defining block:

$$|D| = \max \{|d_1|, |d_2|, \dots, |d_k|\}$$

We assume that agents use their decomposition as a sort of template for generating new solutions, through the following trial-and-error process: a block of the decomposition is randomly chosen, a new sub-configuration for this block is (randomly) generated by mutating at least one and up to all elements of this block but holding the rest of the configuration (outside the considered block) unchanged. If the newly generated configuration is preferred to the former then it is kept, otherwise it is discarded.

More precisely, let us assume that the current configuration is x^i and take block d_h with its current block-configuration $x^i(d_h)$. Let us now consider a new configuration $x^j(d_h)$ for the same block, if:

$$x^j(d_h) \vee x^i(d_{-h}) \succeq x^i(d_h) \vee x^i(d_{-h})$$

then $x^j(d_h)$ is selected and the new configuration $x^j(d_h) \vee x^i(d_{-h})$ is kept in place of x^i , otherwise $x^j(d_h)$ is discarded and x^i is kept.

6 Individual problem-solving with complete representations

Given an encoding $\Omega : O \mapsto X$ and a decomposition scheme $D = \{d_1, d_2, \dots, d_k\}$, we say that a configuration x^i is a preferred neighbor or simply a **neighbor** of

configuration x^j with respect to a block $d_h \in D$ if the following three conditions hold:

1. $x^i \succeq x^j$
2. $x_\nu^i = x_\nu^j \forall \nu \notin d_h$
3. $x^i \neq x^j$

Conditions 2 and 3 require that the two configurations differ only by components which belong to block d_h . According to the definition, a neighbor can be reached from a given configuration through the trial-and-error process described above.

We call $H_i(x, d_i)$ the set of neighbors of a configuration x for block d_i .

The set of **best neighbors** $B_i(x, d_i) \subseteq H_i(x, d_i)$ of a configuration x for block d_i is the set of the most preferred configurations in the set of neighbors:

$$B_i(x, d_i) = \{y \in H_i(x, d_i) \text{ such that } y \succeq z \forall z \in H_i(x, d_i)\}$$

By extension from single blocks to entire decomposition schemes, we can give the following definition of the set of neighbors for a decomposition scheme as:

$$H(x, D) = \bigcup_{i=1}^k H_i(x, d_i)$$

A configuration is a **local optimum** for the decomposition scheme D if there does not exist a configuration y such that $y \in H(x, D)$.

Given an encoding $\Omega : O \mapsto X$ and a decomposition scheme $D = \{d_1, d_2, \dots, d_k\}$ which characterize and agent's representation, its set of local optima is fully determined. We call $\Lambda_i(\Omega_i, D_i)$ agent i 's set of local optima.

A first question which arises is the size of such a set Λ_i . If we assume for simplicity that the “real” problem has only one globally optimum solution o_0 , we ask under which condition its encoding $x(o_0)$ is also the only element in Λ_i . The following three statements (rather trivial to prove⁵) provide the basic results on the roles of encodings and decompositions:

1. for any complete encoding Ω the degenerate decomposition (i.e. no decomposition at all) $D = \{1, 2, \dots, N\}$ guarantees that $\Lambda = x(o_0)$, i.e. that the global optimum is also a unique local optimum;
2. for any problem there exist encodings for which $D = \{1, 2, \dots, N\}$ is also the only decomposition having such a property;

⁵For a more precise formal argument, proofs and extensions, see (Marengo & Dosi 2005).

3. for any problem there exist a complete encoding such that also the finest decomposition $D = \{\{1\}, \{2\}, \dots, \{N\}\}$ (and, on that respect, any other decomposition) guarantees that $\Lambda = x(o_0)$, i.e. that the global optimum is also a unique local optimum;

Taken together these three propositions have the following implications. First, in a sense the encoding is much more powerful than the decomposition in determining the difficulty of a problem: by appropriately acting on the encoding any finite problem can be made optimally solvable by any decomposition including the finest one. Note that if the finest decomposition is used that optimal solution is found in linear time with the simplest possible blind trial-and-error algorithm which changes one dimension at a time without caring for possible interdependencies. Second, if the encoding is not optimal in this sense, then it is possible that only the degenerate decomposition allows to locate with certainty the global optimum, while finer decompositions make the set Λ of local optima grow larger and larger. Note however that the degenerate decomposition, i.e. not decomposing the problem at all, is a very inefficient and time consuming search algorithm as it requires a search time exponential in N . In general finer decompositions will lead the search process to suboptimal solutions but do so relatively quickly and therefore may display higher efficiency with respect to coarser decompositions which certainly locate better solutions but more slowly: there is a trade-off between optimality of the outcome and speed of search (Marengo & Dosi 2005).

7 Institutional analysis

In the previous section we have built a model of individual problem-solving which suggests that only under special conditions will individual agents solve the problem optimally and efficiently (i.e. minimizing time and cost of search). Now we ask whether a collection of heterogenous agents, that is characterized by diverse encodings and decompositions, can perform better, worse or equally than individual agents. We will perform this analysis by making precise hypothesis on the restrictions which property rules can put upon the use and sharing of solutions found by individual agents. In particular we will focus upon two issues, namely:

1. is the use of a current solution vetoed or limited for activities of “improving around” in order to find more valuable solutions?
2. are such restrictions – if they exist – defined only on entire solutions or also on smaller modules? (rights fragmentation)

We begin by analyzing the latter question.

7.1 Rights' fragmentation

Our model of problem-solving adumbrates delicate trade-offs between decomposability, complexity reduction and search speed on the one hand and asymptotic optimality on the other.

Let us consider the trade-off between speed of search and optimality. For instance, Figure 1 shows that in Kauffman's NK random landscapes⁶ there exist only decomposition schemes of size N or just below N even for very small values of K (that is, for highly correlated landscapes). In other words, a little bit of interdependence spread across the set of components immediately makes a system practically indecomposable.

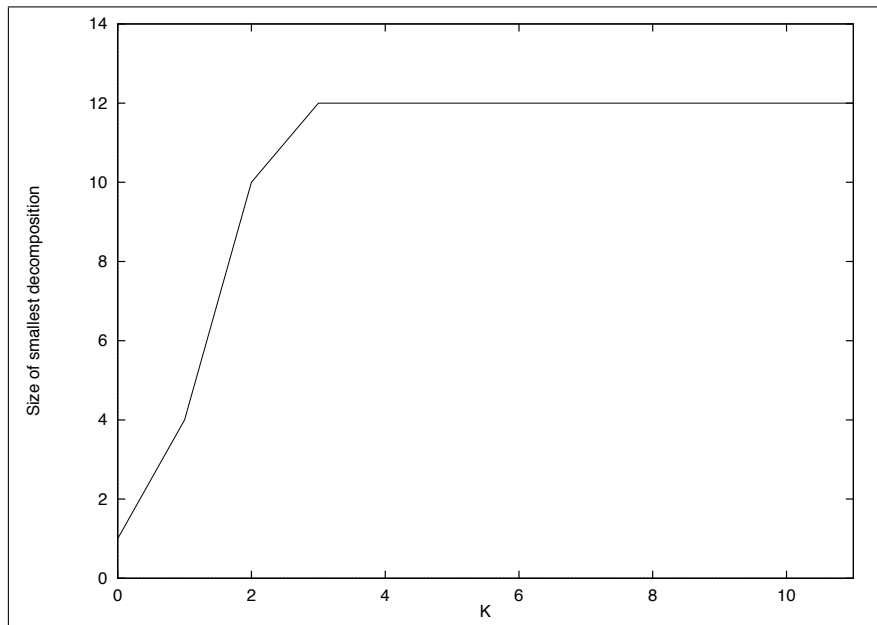


Figure 1: Size of minimum decomposition schemes for random NK problems. ($N=12$)

We can soften the perfect decomposability requirement into one of near-decomposability; we no longer require the problem to be decomposed into completely separated sub-problems (i.e. sub-problems that fully contain all interdependencies) but we might be content to find sub-problems that contain

⁶An NK random fitness landscape is similar to our definition of “problem” except that, instead of a preference relation, a real valued fitness function $F : X \mapsto \mathbb{R}$ is defined as an average of each component’s fitness contribution. The latter is a random realization of a random variable uniformly distributed over the interval $[0, 1]$ for each possible configuration of the K -size block of the other components with which each component interacts (Kauffman 1993). Note, however, that Kauffman’s K is not a good ex-post complexity measure (in terms of its decomposability) of the optimization problem on the resulting fitness landscape; small values of K usually generate landscapes that are not decomposable, but on the other hand, it is always possible that, even with very high values of K , the resulting landscape is highly decomposable.

the most “relevant” interdependencies, while less relevant ones can persist across sub-problems. In this way, optimizing each sub-problem independently will not necessarily lead to the global optimum, but to a “good” solution.

Figure 2 shows, for randomly generated problems,⁷ that if second-best solutions are accepted we can reduce considerably the size of the decomposition schemes and thus the expected search time: every reduction of 1 unit of the size of the decomposition scheme halves the corresponding expected search time. This shows that the organizational structure sets a balance in the trade-off between search and adaptation speed and optimality.

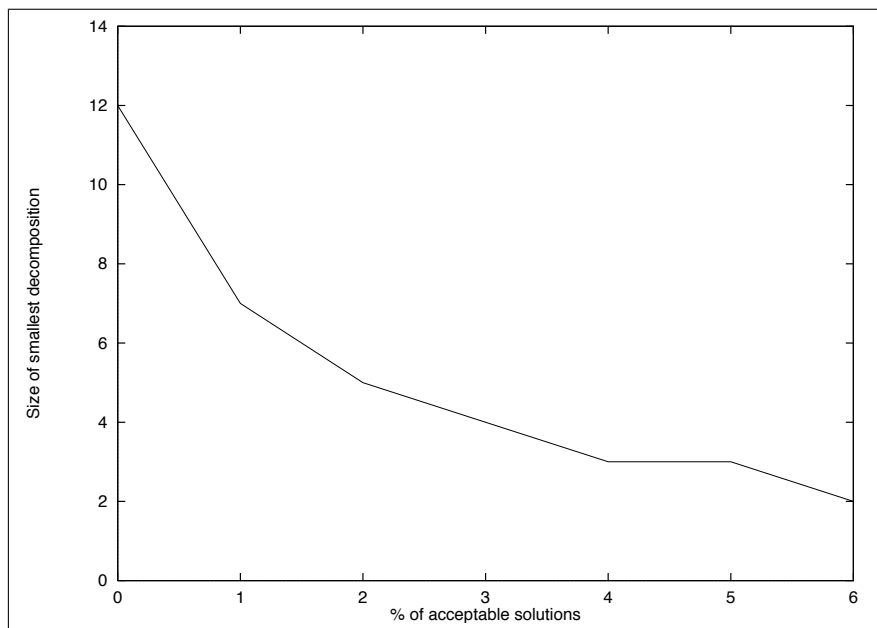


Figure 2: Near decomposability.

It is easy to argue that in complex problem environments, characterized by strong and diffused interdependencies, such a trade-off will tend to produce structures that are more decomposed than what would be optimal given the interdependencies of the problem space. This property is shown in Figures 3 and 4, that present the typical search paths on a non-decomposable problem of two search processes driven, respectively, by decompositions:

$$D1 = \{1, 2, \dots, 12\}$$

$$D12 = \{\{1\}, \{2\}, \dots, \{12\}\}$$

Figure 3 shows the first 180 iterations in which the more decomposed structure (D12) quickly climbs the problem space and outperforms a search based on a coarser decomposition. If there were a tight selection environment, a more-than-optimally-decentralized structure would quickly displace structure D1, that reflects the “true” decomposition of the underlying problem space.

⁷In this figure and the following we indicate on the vertical axis the rank of configurations re-parametrized between 0 (worst) and 1 (best).

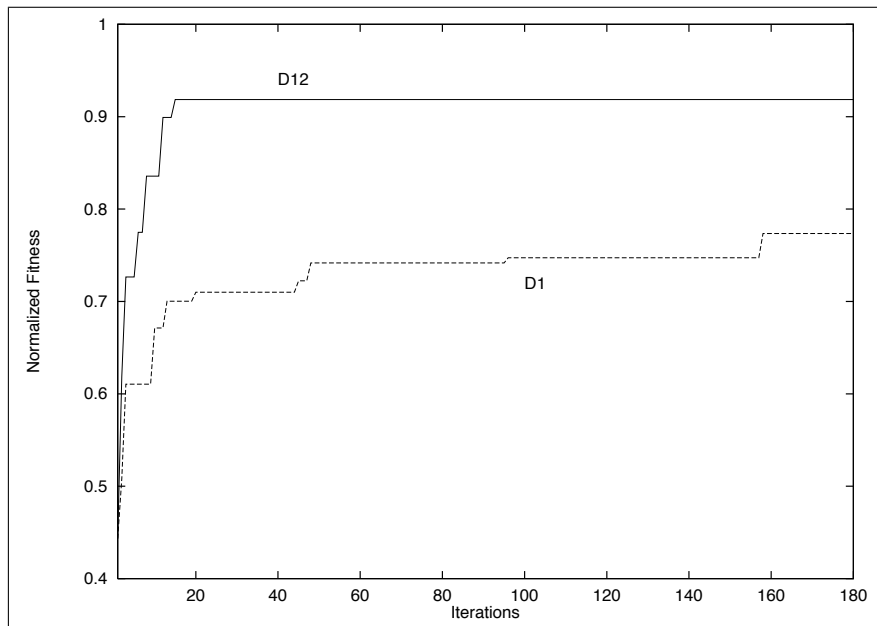


Figure 3: Fitness values for search processes with finest (D12) and coarsest(D1) decompositions (N=12). First 180 iterations...

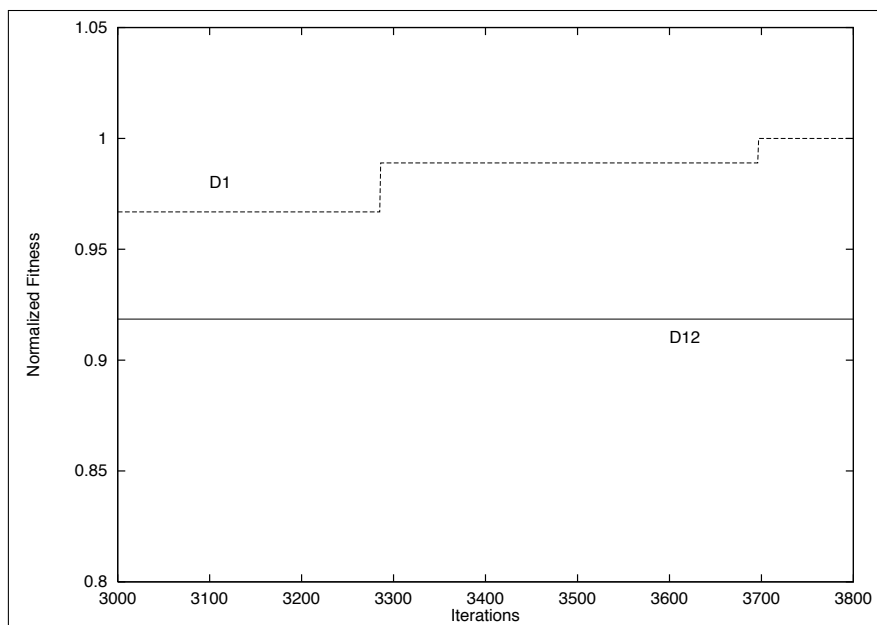


Figure 4: ... after 3000 iterations.

However, the search process based on the finest decomposition quickly reaches a local optimum from where no further improvement can occur, while the process based on the coarser decomposition keeps searching and climbing slowly. Figure 4 shows iterations between 3000 and 3800, where the finest decomposition is still locked-into the local optimum it reached after very few iterations, while the coarsest one slowly reaches the global optimum (normalized to 1). Strong selective pressure therefore tends to favor structures whose degree of decentralization is higher than what would be optimal from a mere problem-solving perspective.

This result is even stronger in problems that we could define as “modular”, those characterized by blocks with strong interdependencies within blocks and much weaker (but non-zero) interdependencies between blocks; in these problems, higher levels of decompositions can be achieved at lower costs in terms of sub-optimality.

All in all we have shown that if the innovation process in software is characterized by high and diffused interdependencies, then it cannot be reduced to a “perfectly Coasian” world. Even lacking transaction costs and with perfectly competitive markets finer property rights are not in general conducive to higher technological efficiency. On the other hand competitive selection environments tend to favor more than optimally decomposed structures, that is more finely defined IPR’s, because of the higher adaptation speed of such structures.

7.2 Proprietary vs. free problem-solving

Consider now a polyarchy (Sah & Stiglitz 1986), in which a collection of problem solving agents can work on the current solution and try to find an improvement, if one agent finds an improvement then this becomes the new current solution and the process begins again, until none of the agents can make any further improvement on the current solution. We can compare three kinds of institutional settings: one in which solutions are free and open-source and once found by an agent become immediately available to everybody else for cumulatively making further improvements. One in which on the contrary solutions are proprietary and are not disclosed and any solution close enough to one already protected is prohibited. Finally one in which existing solutions can be licensed against the payment of a fee.

Consider first the free and open-source environment. To be more precise, we suppose that agents are called sequentially and in a random order to perform their search starting from a current solution. Each agent stop when a local optimum (for its representation) is located and this local optimum becomes the new current solution. Then another agent is called and the process is repeated until no agent can make any further improvement. The outcome is the locally

optimal solution for the polyarchic group⁸.

This process can be described as a path on the set of different agents' local optima Λ_i . It is easy to understand that given enough diversity in such sets, even if none of them is a singleton containing only the global optimum, the polyarchy's set of local optimum that we call Λ_P can itself be such a singleton. Such a diversity can be generated either by diversity of encodings or by diversity of decompositions, but again diversity of encodings seems in a sense more powerful. In fact, the following results are easy to prove:

1. if all agents share the same encoding Ω , but have different decompositions, then it is possible that Λ_P be a singleton, i.e. that the group search process locates with certainty the global optimum;
2. however, if the problem is not decomposable for the common encoding Ω , a necessary condition is that at least one agent holds the degenerate decomposition $D = 1, 2, \dots, N$;
3. if all agents share the same finest decomposition $D = \{\{1\}, \{2\}, \dots, \{N\}\}$ but have diverse encodings, it is possible that Λ_P be a singleton even when none of the Λ_i is a singleton itself.

At the other extreme, if there is no variety among agents, i.e. if they all share the same encoding and decomposition, then there is no difference between group and individual performance, as $\Lambda_P = \Lambda_i$ for a generic agent.

Following (Hong & Page 2001) we can also define a notion of productivity of agents: given a group and a new agent, does the addition of the latter increase, and how much, the group performance? (Hong & Page 2001) show that an agent's productivity is extremely variable in the group's composition and has no significant correlation with the agent's "ability", as measured by the expected value of its local optima.

8 Tentative conclusions

Back in the years in which what was good for FIAT was good for Italy, if an economists' committee would have proposed that the government should support the efforts of volunteers to design and build their own cars for exchange, sale or free distribution, they would have probably been locked into a psychiatric hospital (if Scelba or Tambroni did not get them first).

Yet, something pretty similar to this happened when the Italian government (as well as the UE and the US) recommended in 2001 that Open Source

⁸A somehow similar model has been investigated by (Hong & Page 1998) and (Hong & Page 2001).

Software be supported and adopted in public offices as a strategic national choice.

What is surprising, of course, is that something can be observed in a central locus of contemporary economies that simply should not be there: production, exchange and distribution outside the realm of both market interactions and firm based hierarchies and out of the control of traditional property rights as the main incentive mechanism.

In broader terms, what is really noteworthy is how in software, computer communication and even in biotechnologies, firm strategies that do not depend on strong IP and forms of division of labor that are not strictly based on hierarchies have often outperformed “traditional” appropriation strategies and divisions of labor. Examples are the adoption of such a standard as TCP/IP rather than CompuServe, Prodigy or MSN or the strategy adopted by Merck in pursuing complementary DNA sequencing (case discussed in (Heisenberg 1996)).

In our opinion the theme for discussion and analysis here is not the mere fact that Open Source has emerged as an alternative mode of production and distribution but rather the relationship between division of labor, technology and protection regimes. Understanding causal relations between technology and institutions is an old problem but, surprisingly enough, the question of how IP regimes and the characteristic of information technology itself affect information production has remained relatively in the background of the current debate.

To paint an old question with an extremely broad brush, the origins of firm based hierarchies can be described along two basic dimensions. The first runs as: what caused them to emerge? The second can be described by the question: what is their essential nature?

In his famous 1974 paper, S. Marglin (Marglin 1974) – somehow rejecting a straightforward marxian position – set forth the argument that the real engine of exploitation is organization rather than technology. Task fragmentation was the main tool used in order to deskill work and render it so simple and undifferentiated that untrained proletariat could replace skilled artisans. According to this perspective, the division of labor serves not an efficiency ideal but it rather gives a capitalist the way to more efficiently control workers.

On the other hand Williamson and North, while subscribing the view that hierarchy based firms did emerge for a matter of organization, focus on efficiency considerations as to why they did emerge. In this sense, they largely describe firms in terms of minimizing transaction costs, costs related to coordinating finely subdivided processes and costs of monitoring product quality.

Given these considerations, we ask two questions. First, defining peer production as a possible organization of a productive process, our question is: why

did peer production (at least in some relevant cases) outperform market based and hierarchy based production in (at least some) information production activities? Given our considerations on the increased relevance of human capital in information production, the relative lowering of physical capital importance and the declining costs of communication, we conjecture that peer production has a comparative advantage in acquiring and processing information about human capital available to contribute to information production projects.

It is noteworthy that as far as 14th century (and till the beginning of 20th century), the attitude towards patenting was to award domestic patents in order to transfer technology produced and developed abroad for domestic usage. So patenting was first and foremost serving its disclosure function rather than its protecting one.

Originally, the IP system was designed with an eye to industrial applications as far as they were concerned with the physical transformations of raw materials and not with digital goods. In particular, patenting requires certain standards to be met (non obviousness, novelty, applicability). In this perspective, an algorithm cannot be patented since, as such and per se, it does not have any direct industrial applicability.

Copyright, quite on the contrary, aims at protecting the expression of ideas in some kind of medium. Under the Berne convention, no precise formal standards are to be met in order to copyright something. Copyright applies not to raw ideas (which are considered to be part of nature and thus not created) but solely to their expression.

According to this reading, no algorithm can be copyrighted as no algorithm in itself (before software implementation) can be expressed with, say, artist-specific flair and every algorithm is part of some underlying state of nature.

So, as paradoxical as it may seem, when an algorithm is copyrighted the most relevant part of it, namely the underlying idea, is left unprotected. What an IP owner does is acting on the implementation or on the expression's flair and it is on that dimension that producers try to distinguish themselves (e.g. controls, menus...). Of course this kind of competition is wasteful, costly in resources and it does nothing to improve both the availability and the quantity of digital goods for society.

Also note incidentally that being the term of protection the author's life plus 70 years (or 95 years in total), by the time a piece of software falls in the public domain there will be no machine that can possibly run it. So, the term of copyright is actually unlimited.

Also note that when applied to software the system of copyright protect software (ex-ante incentive) without creating any new knowledge in return nor diffusing knowledge in society (ex-post efficiency). Consider, for instance, that when the copyright system protects Melville at least society can appreciate

and see how he wrote (his literary style, his way of building plots....) and many people can improve their own writing style by reading Moby Dick and receiving inspiration and plenty of usable example of better narrative style. As to software, this ceases to be true: once a piece of code is compiled what you get is something which is totally unreadable. Notwithstanding this fact, in order to copyright a piece of code the author does not have to reveal the source code.

In the long run, this way of protecting knowledge does nothing but destroying knowledge.

These ideas seem to be very much in line with (Boldrin & Levine 2002) arguments. They maintain that “property” for e.g. land has nothing to do with “property” in Intellectual Property and that the same notion does not apply to ideas as it does to land. According to their perspective, the view that equates the two meanings stems from confusing the abstract notion of “idea” with the concrete implementation or embodiment of an idea (which, as we have seen, is the key problem in copyrighting algorithms: i.e. the idea vs. the flair/expression quibble).

According to Boldrin and Levine, IP law has come to mean not just the right to sell and own ideas but the right to regulate their use. IP law has two basic components: the first is the right to own and sell ideas (right of first sale) while the second is the right to control their use after sale (downstream licensing). The key point in their argument is that the traditional analysis leading to the necessity of downstream licensing is critically based on assuming that costs in innovative activity are fixed costs. On the contrary, these are sunk costs (i.e. costs related to producing the first unit). Sunk costs do not pose any particular problem nor serious threat to competition. As a matter of facts no one claims that a monopoly right should be legally accorded to producers of any good whose production involves sunk costs.

References

- Baldwin, C. Y. & Clark, K. B. (2000), *Design Rules. The Power of Modularity*, MIT Press, Cambridge Mass.
- Boldrin, M. & Levine, D. K. (2002), ‘The case against intellectual property’, *American Economic Review (Papers and Proceedings)* **92**, 209–212.
- Coase, R. H. (1960), ‘The problem of social cost’, *Journal of Law and Economics* **3**, 1–44.

- David, P. A. (1992), Knowledge, property, and the system dynamics of technological change, *in* ‘Proceedings of the World Bank Annual Conference on Development Economics’, pp. 215–248.
- Heisenberg, R. (1996), Intellectual property at the public-private divide: the case of large-scale dna sequencing, Working paper, Michigan University Law School, Ann Arbor.
- Hong, L. & Page, S. E. (1998), Diversity and optimality, Unpublished Manuscript.
- Hong, L. & Page, S. E. (2001), ‘Problem solving by heterogeneous agents’, *Journal of Economic Theory* **97**, 123–163.
- Kauffman, S. A. (1993), *The Origins of Order*, Oxford University Press, Oxford.
- Marengo, L. & Dosi, G. (2005), ‘Division of labor, organizational coordination and market mechanisms in collective problem-solving’, *Journal of Economic Behavior and Organization* p. in press.
- Marengo, L., Pasquali, C. & Valente, M. (2005), Decomposability and modularity of economic interactions, *in* W. Callebaut & D. Rasskin-Gutman, eds, ‘Modularity: Understanding the Development and Evolution of Complex Natural Systems’, MIT Press, Cambridge, MA, pp. 835–897. Vienna Series in Theoretical Biology.
- Marglin, S. A. (1974), ‘What do bosses do? the origins and function of hierarchy in capitalist production’, *Review of Radical Political Economics* **6**, 33–60.
- Milgrom, P. & Roberts, J. (1990), ‘The economics of modern manufacturing’, *American Economic Review* **80**, 511–528.
- Quah, D. (2003), Digital goods and the new economy, CEPR Discussion Papers Series 3846, Centre for Economic Policy Research, London.
- Sah, R. K. & Stiglitz, J. E. (1986), ‘The architecture of economic systems: hierarchies vs. polyarchies’, *American Economic Review* **76**, 716–727.
- Simon, H. A. (1969), *The Sciences of the Artificial*, MIT Press, Cambridge, MA.
- Simon, H. A. (2002), ‘Near decomposability and the speed of evolution’, *Industrial and Corporate Change* **11**, 587–599.