

PROGETTO DI UNA UNITÀ DI RICERCA - MODELLO B  
Anno 2005 - prot. 2005015491\_001

**PARTE I**

**1.1 Programma di Ricerca afferente a**

*Area Scientifico Disciplinare 01: Scienze matematiche e informatiche 100%*

---

**1.2 Durata del Programma di Ricerca**

*24 Mesi*

---

**1.3 Coordinatore Scientifico del Programma di Ricerca**

**ROSSI**                      **FRANCESCA**                      *frossi@math.unipd.it*

*INF/01 - Informatica*

*Università degli Studi di PADOVA*

*Facoltà di SCIENZE MATEMATICHE FISICHE e NATURALI*

*Dipartimento di MATEMATICA PURA ED APPLICATA*

---

**1.4 Responsabile Scientifico dell'Unità di Ricerca**

**ROSSI**                                      **FRANCESCA**  
*Professore Ordinario*                      *07/12/1962*                      *RSSFNC62T47A271Z*

*INF/01 - Informatica*

*Università degli Studi di PADOVA*

*Facoltà di SCIENZE MATEMATICHE FISICHE e NATURALI*

*Dipartimento di MATEMATICA PURA ED APPLICATA*

*049/8275982*                                      *049/8275892*                                      *frossi@math.unipd.it*  
*(Prefisso e telefono)*                                      *(Numero fax)*                                      *(Indirizzo posta elettronica)*

---

**1.5 Curriculum scientifico del Responsabile Scientifico dell'Unità di Ricerca**

**Testo italiano**

*Francesca Rossi e' professore ordinario di Informatica presso il Dipartimento di Matematica Pura ed Applicata dell'Universita' di Padova da Novembre 1998. Precedentemente (dal 1998) era professore asociato presso lo stesso dipartimento, e (dal 1992) era ricercatore presso il Dipartimento di Informatica dell'Universita' di Pisa, dove ha anche ottenuto il suo dottorato di ricerca in Informatica nel 1993. Ha passato diversi periodi all'estero, tra cui 18 mesi all'MCC (Austin, TX), e vari periodi brevi alla NEC (Princeton, NJ), al Weizmann Institute (Rehovot, Israele), ai Bell Communication Research Labs (NJ, USA), e a Xerox PARC (Menlo Park, CA, USA).*

*I suoi interessi di ricerca vanno dall'Intelligenza Artificiale ai Linguaggi di Programmazione, con particolare attenzione alla*

programmazione con vincoli, agli algoritmi di soluzione di vincoli, e ai vincoli soft. Ha anche lavorato su semantica, grammatiche per grafi, programmazione logica, reti di Petri.

Ha organizzato workshops relativi alla sua area di ricerca, nel 1998 e' stata conferenze chair della conferenza internazionale su vincoli e nel 2003 program chair della stessa conferenza. E' presidente dell'Executive Committee della associazione internazionale della programmazione con vincoli. E' coordinatore dell'area "Programmazione con vincoli" della rete europea su "Logica Computazionale". Ha partecipato ai comitati di programma di molte conferenze internazionali di Intelligenza Artificiale e Programmazione con Vincoli. E' membro del comitato editoriale della rivista *Constraints* e del *Journal of Artificial Intelligence Research*.

Ha pubblicato circa 80 articoli, tra riviste internazionali, libri, atti di conferenze o workshop internazionali con revisione, e 8 edizioni di volumi.

Il ruolo che svolgera' in questo progetto di ricerca e' sia di coordinamento delle linee di ricerca, che anche di effettiva ricerca teorica e sperimentale su alcune delle linee di ricerca del progetto. In particolare, la sua attivita' di ricerca si focalizzera' soprattutto sulle linee relative ai vincoli soft.

### **Testo inglese**

Francesca Rossi is full professor of Computer Science at the Department of Pure and Applied Mathematics of the University of Padova since March 2001. Previously she has been associate professor in the same department since Nov. 1998, and assistant professor at the Computer Science Department of the University of Pisa (since 1992), where she also obtained her Ph.D. in Computer Science in 1993. She has spent several visiting periods abroad, among which 18 months at MCC (Austin, TX), and various short periods at NEC (Princeton, NJ), at the Weizmann Institute (Rehovot, Israel), at Bell Communication Labs (NJ), and at Xerox PARC (Menlo Park, CA).

Her research interests range from Artificial Intelligence to Programming Languages, with particular attention to constraint programming, constraint solution algorithms, and soft constraints. She has also worked on language semantics, graph grammars, logic programming, and Petri nets.

She belongs to the editorial board of the journal "Constraints" by Kluwer and she is associate editor of JAIR (Journal of AI research). She has organized several workshops, in 1998 she has been the conference chair of the international conference on constraint programming and in 2003 she has been program chair of the same conference. Moreover, she is the president of the executive committee of the constraint programming association. She is also the coordinator of the constraint logic programming area of the european network Compulog Net. She has participated in several program committees of many international conferences of Artificial Intelligence and Constraint Programming.

She has published about 80 articles, among international journals, books, proceedings of international conferences or workshops, and 8 edited volumes.

Her role in this project is to coordinate the research units, to supervise the research in her own unit, and also to work actively on several research lines of the project. In particular, her research activity will be focussed mainly on soft constraints.

---

## **1.6 Pubblicazioni scientifiche più significative del Responsabile Scientifico dell'Unità di Ricerca**

1. F. ROSSI, K. B. VENABLE, T. WALSH (2005). *Aggregating preference cannot be fair*. *INTELLIGENZA ARTIFICIALE*. vol. 10 pp. 50-60 ISSN: 1724-8035
2. F. ROSSI, A. SPERDUTI (2004). *Acquiring both constraint and solution preferences in interactive constraint systems*. *CONSTRAINTS*. vol. 9 pp. 311-332 ISSN: 1383-7133
3. F. ROSSI, T. FRUEHWIRTH, M. MARTE AND F. ROSSI (2004). *Soft Constraint Propagation and Solving in Constraint Handling Rules*. *COMPUTATIONAL INTELLIGENCE*. vol. 20 pp. 287-307 ISSN: 0824-7935
4. M. S. FRANZIN, E. C. FREUDER, F. ROSSI, R. WALLACE (2004). *Multi-agent meeting scheduling with preferences: efficiency, privacy loss, and solution quality*. *COMPUTATIONAL INTELLIGENCE*. vol. v. 20 (n. 2) pp. 264-286 ISSN: 0824-7935
5. STEFANO BISTARELLI, ROSELLA GENNARI, F. ROSSI (2003). *General Properties and Termination Conditions for Soft Constraint Propagation*. *CONSTRAINTS*. vol. 8 (1) pp. 79-97 ISSN: 1383-7133
6. S. BISTARELLI, P. CODOGNET, F. ROSSI (2002). *Abstracting Soft Constraints: Framework, Properties, Examples*. *ARTIFICIAL INTELLIGENCE*. vol. 139 pp. 175-211 ISSN: 0004-3702 Issue 2.
7. S. BISTARELLI, U. MONTANARI, F. ROSSI (2002). *Soft Constraint Logic Programming and Generalized Shortest Path Problems*. *JOURNAL OF HEURISTICS*. vol. 8 pp. 25-41 ISSN: 1381-1231
8. R. BRUNI, U. MONTANARI, F. ROSSI (2001). *An interactive semantics of logic programming*. *THEORY AND PRACTICE OF LOGIC PROGRAMMING*. vol. 1 pp. 647-690 ISSN: 1471-0684
9. BUENO F., HERMENEGILDO M., MONTANARI U., F. ROSSI (1998). *Partial Order and Contextual Net Semantics for Atomic and Locally Atomic CC Programs*. *SCIENCE OF COMPUTER PROGRAMMING*. vol. 30 ISSN: 0167-6423
10. F. ROSSI, A. Sperduti (1998). *Lerning Solution preferences in constraint problems*. *JOURNAL OF EXPERIMENTAL & THEORETICAL ARTIFICIAL INTELLIGENCE*. vol. 10 ISSN: 0952-813X
11. U. Montanari and F. Rossi, F. ROSSI (1998). *Graph Rewriting, Constraint Solving and Tiles for Coordinating Distributed Systems*. *APPLIED CATEGORICAL STRUCTURES*. ISSN: 0927-2852

12. BISTARELLI S., MONTANARI U., F. ROSSI (1997). *Semiring-based Constraint Solving and Optimization*. JOURNAL OF THE ACM. vol. 44, 2
13. MONTANARI U., F. ROSSI (1993). *Graph rewriting for a partial ordering semantics of concurrent constraint programming*. THEORETICAL COMPUTER SCIENCE. vol. 109 ISSN: 0304-3975
14. MONTANARI U., F. ROSSI (1991). *Constraint Relaxation May be Perfect*. ARTIFICIAL INTELLIGENCE. vol. 48 pp. 143-170 ISSN: 0004-3702

## 1.7 Risorse umane impegnabili nel Programma dell'Unità di Ricerca

### 1.7.1 Personale universitario dell'Università sede dell'Unità di Ricerca

#### Personale docente

n°	Cognome	Nome	Dipartimento	Qualifica	Settore Disc.	Mesi Uomo	
						1° anno	2° anno
1.	ROSSI	Francesca	Dip. MATEMATICA PURA ED APPLICATA	Prof. Ordinario	INF/01	7	7
<b>TOTALE</b>						<b>7</b>	<b>7</b>

#### Altro personale

n°	Cognome	Nome	Dipartimento	Qualifica	Mesi Uomo		
					1° anno	2° anno	
1.	Venable	Kristen Brent	Dip. MATEMATICA PURA ED APPLICATA	collaboratore	6	6	
<b>TOTALE</b>						<b>6</b>	<b>6</b>

### 1.7.2 Personale universitario di altre Università

#### Personale docente

n°	Cognome	Nome	Università	Dipartimento	Qualifica	Settore Disc.	Mesi Uomo	
							1° anno	2° anno
1.	GAVANELLI	Marco	FERRARA	Dip. INGEGNERIA	RU	ING-INF/05	4	4
<b>TOTALE</b>							<b>4</b>	<b>4</b>

#### Altro personale

Nessuno

### 1.7.3 Titolari di assegni di ricerca

Nessuno

### 1.7.4 Titolari di borse

n°	Cognome	Nome	Dipartimento	Anno di inizio borsa	Durata(in anni)	Tipologia	Mesi Uomo	
							1° anno	2° anno
1.	Pini	Maria Silvia	Dip. MATEMATICA PURA ED APPLICATA	2004	3	Dottorato	6	6
<b>TOTALE</b>							<b>6</b>	<b>6</b>

**1.7.5 Personale a contratto da destinare a questo specifico programma**

n° Qualifica	Costo previsto	Mesi Uomo		Note
		1° anno	2° anno	
1. <i>Borsista</i>	9.000	4	4	
<b>TOTALE</b>	<b>9.000</b>	<b>4</b>	<b>4</b>	

**1.7.6 Personale extrauniversitario indipendente o dipendente da altri Enti**

n°	Cognome	Nome	Nome dell'ente	Qualifica	Mesi Uomo	
					1° anno	2° anno
1.	<i>Cesta</i>	<i>Amedeo</i>	<i>CNR - ISTC (Istituto per le Scienze e le Tecnologie Cognitive)</i>	<i>ricercatore</i>	2	2
2.	<i>Oddi</i>	<i>Angelo</i>	<i>CNR - ISTC</i>	<i>ricercatore</i>	2	2
3.	<i>Policella</i>	<i>Nicola</i>	<i>CNR-ISTC</i>	<i>assegnista</i>	3	3
	<b>TOTALE</b>				<b>7</b>	<b>7</b>

## **PARTE II**

### **2.1 Titolo specifico del programma svolto dall'Unità di Ricerca**

#### **Testo italiano**

*Vincoli e preferenze per problemi di scheduling e sistemi multi agente o con criteri multipli.*

#### **Testo inglese**

*Constraints and preferences in scheduling problems and multi-agent or multi-criteria systems.*

---

### **2.2 Settori scientifico-disciplinari interessati dal Programma di Ricerca**

*INF/01 - Informatica*

---

### **2.3 Parole chiave**

#### **Testo italiano**

*PROGRAMMAZIONE CON VINCOLI ; PREFERENZE ; RICERCA OPERATIVA ; PROBLEMI DI SCHEDULAZIONE ; SISTEMI MULTI-AGENTE ; SISTEMI A CRITERI MULTIPLI*

#### **Testo inglese**

*CONSTRAINT PROGRAMMING ; PREFERENCES ; OPERATION RESEARCH ; SCHEDULING PROBLEMS ; MULTI-AGENT SYSTEMS ; MULTI-CRITERIA SYSTEMS*

---

### **2.4 Base di partenza scientifica nazionale o internazionale**

#### **Testo italiano**

*La programmazione con vincoli:*

*La Programmazione con Vincoli [1, 19] e' una metodologia di programmazione che permette di formalizzare il problema da risolvere in modo puramente dichiarativo con l'ausilio di formule logiche dette appunto vincoli. La fase di computazione e' tipicamente basata sulla ricerca di una o piu' soluzioni che soddisfino l'insieme di vincoli imposti. La Programmazione con Vincoli permette di tenere ben distinte la fase di formalizzazione da quella di ricerca delle soluzioni. Tuttavia anche questa seconda fase viene guidata dai vincoli che devono essere rispettati, al fine di ridurre sensibilmente lo spazio di ricerca.*

*Un risolutore di vincoli e' una procedura effettiva in grado di determinare la soddisfaccibilita' di vincoli e di trovare soluzioni che li soddisfino tutti, se esistono. Diversi risolutori sono stati proposti dagli anni '80 in poi, per vari domini di vincoli e con proprieta' e tecniche implementative diverse.*

*L'utilizzo di vincoli su domini finiti permette di formulare problemi di ottimizzazione combinatoria, quali problemi di scheduling, timetabling, etc. [1, 19]. I risolutori di vincoli per tali domini sono basati sul concetto di propagazione di vincoli, utilizzando tipicamente la tecnica di arc consistency [17,18] o sue semplificazioni, e sulla ricerca nello spazio delle soluzioni, controllata dalla propagazione dei vincoli che permette di rilevare assegnamenti parziali delle variabili che non possono condurre a soluzioni accettabili. Per problemi di ottimizzazione, la ben nota tecnica di branch-and-bound collabora con la propagazione di vincoli per diminuire il numero di soluzioni da investigare.*

*I vincoli soft:*

*In molti problemi reali, non tutte le soluzioni sono ugualmente desiderate. Inoltre, in alcune situazioni i vincoli sono naturalmente associati a delle preferenze che rappresentano la gravita' della loro violazione. Per poter modellare fedelmente queste situazioni, il formalismo classico dei vincoli e' stato generalizzato per permettere la descrizione di preferenze. Le preferenze non sono altro che dei vincoli nei quali, per ogni combinazione di valori alle variabili del vincolo, viene associato un elemento (la preferenza per quella combinazione) da un insieme ordinato (totalmente o parzialmente).*

*Negli ultimi 10 anni sono nati vari formalismi per modellare e gestire problemi con preferenze, tra cui i piu' generali sono il formalismo basato sui semianelli [3,4], quello dei valued CSP [24], e le cosiddette CP-nets [5].*

*Risolvere un problema con preferenze e' in genere piu' complesso che risolvere un problema di vincoli, in quanto ora le soluzioni volute sono solo quelle ottime rispetto alle preferenze. Tecniche di branch and bound, di propagazione di vincoli estesa alle preferenze, e di astrazione, sono utili per sviluppare un risolutore per problemi con preferenze.*

*Noi vogliamo sviluppare nuovi formalismi per modellare preferenze piu' complesse di quelle gestite finora, ad esempio preferenze bipolari [9], che permettono di descrivere sia desideri che soluzioni da evitare, o anche preferenze sia qualitative che quantitative.*

Vogliamo anche poter modellare ed aggregare preferenze di più agenti.

*Preferenze e incertezza:*

Oltre alle preferenze, molti problemi reali presentano parti incerte, in cui l'utente non può decidere niente ma solo aspettare che le cose avvengano. Per questo è importante poter gestire l'incertezza in problemi con vincoli e/o preferenze. Lo scopo è allora trovare una o più soluzioni che siano buone secondo le preferenze ma anche sufficientemente compatibili con l'incertezza del problema.

Vari formalismi per modellare l'incertezza sono stati usati, a seconda del tipo di incertezza presa in considerazione. Ad esempio, si può pensare di associare ad ogni evento incerto una probabilità, o, in mancanza di una stima, un grado di possibilità [10,27].

Alcuni lavori hanno già gestito alcuni tipi di preferenze con incertezza [21]. Noi vogliamo estendere questi approcci e generalizzare a qualunque tipo di preferenze.

*Vincoli e scheduling:*

Problemi di scheduling emergono in diverse situazioni reali [2,15,20,23]. Infatti, tale problema può essere definito come l'assegnamento di tempi di inizio e di fine per un insieme di attività soggette a certi vincoli. I vincoli tipicamente sono di tipo temporale o di disponibilità di risorse. La coordinazione della produzione in una azienda, la gestione di missioni spaziali, la schedulazione di trasporti sono esempi rappresentativi di problemi di scheduling. Oltre ai vincoli, di solito ci sono anche obiettivi e preferenze da ottimizzare. Quindi un problema di scheduling richiede di capire quando eseguire certe attività in modo che la soluzione finale garantisca buone performance.

In tale ambito, il nostro obiettivo è di sviluppare tecniche capaci di considerare problemi con preferenze sulle attività e/o di ottenere soluzioni che siano robuste a modifiche. Per quanto riguarda il primo punto, si vuole introdurre il concetto di preferenza all'interno di architetture state dell'arte (OOSCAR [6]) così come negli associati metodi risolutivi [7,8]. Nella produzione di soluzioni robuste si vuole perseguire l'idea di produrre soluzioni flessibili [22] estendendo il concetto di flessibilità di risorse e sviluppando un simulatore per verificare la validità della combinazione della robustezza della soluzione iniziale con i metodi utilizzati per ri-parla quando necessario.

*Vincoli e ottimizzazione multi-obiettivo:*

L'ottimizzazione multi-obiettivo è considerata fondamentale in moltissime applicazioni reali. Tipicamente, nei problemi di ottimizzazione combinatoria presenti nel mondo industriale, è necessario ottimizzare più di un aspetto, in quanto gli obiettivi sono molteplici e non è sempre possibile raggrupparli in una sola funzione obiettivo. Spesso, poi, gli obiettivi sono di scordanti, per cui una soluzione ottima per una certa funzione obiettivo può non esserlo per le altre.

Il campo è vastissimo ed è stato affrontato in diverse comunità scientifiche: in Intelligenza Artificiale, ed, in particolare, in programmazione a vincoli [13], in ricerca operativa [11], utilizzando sia risolutori specifici (ad esempio, per problemi di scheduling [16,26]) sia risolutori di programmazione lineare (intera) [25], sia tecniche di ricerca locale [14], sia algoritmi evolutivi [12] [28].

Il problema è studiato da tempo e varie tecniche sono state proposte. Un primo metodo consiste nel combinare le diverse funzioni obiettivo in una sola ed utilizzare i classici metodi di ottimizzazione mono-obiettivo per ottenere una sola soluzione ottima. Tutti questi metodi hanno in comune l'idea di trasformare l'ordinamento parziale fra le possibili soluzioni in un ordinamento totale, con diverse caratteristiche in termini di bilanciamento fra le funzioni obiettivo, facilità di ottenere la soluzione (incluso la capacità di potare l'albero di ricerca tramite rilassamenti), necessità di inserire a priori dei fattori per normalizzare le funzioni obiettivo. Sebbene questi metodi siano, in genere, molto efficienti (in quanto richiedono una sola fase di ottimizzazione che riusa componenti già sviluppati), non sempre riescono a cogliere quale sia la soluzione che risulta preferibile per l'utente.

Un'ulteriore classe di algoritmi cerca di fornire all'utente tutto l'insieme delle soluzioni non-domite, cioè delle possibili soluzioni che non sono migliorabili per una funzione obiettivo senza peggiorarne un'altra. L'insieme delle soluzioni non-domite viene chiamato anche frontiera Pareto ottima. Chiaramente, la frontiera Pareto ottima fornisce il massimo di informazioni all'utente, ma è anche meno efficiente, in termini sia di tempo di calcolo che di occupazione di memoria, rispetto ai metodi che propongono una soluzione sola. Il vantaggio principale è che l'utente ha la possibilità di scegliere autonomamente una soluzione a posteriori e non deve accettare a priori quella fornita dall'algoritmo di ottimizzazione.

In questo progetto, si desiderano studiare ed integrare tecniche di Intelligenza Artificiale e di Ricerca Operativa per l'ottimizzazione multi-obiettivo. Lo scopo è quello di sviluppare algoritmi ibridi in grado di generare efficientemente la frontiera Pareto ottima.

### **Testo inglese**

*Constraint programming:*

Constraint programming [1, 19] is a programming methodology which allows one to formalize the problem to solve in a purely declarative way, with the help of logical formulae called constraints. The computation phase is typically based on the search for one or more solutions which satisfy the set of all constraints. Constraint programming allows one to keep the formalization phase and the search for solutions completely separate. However, also the search for solutions is guided by the constraints to be satisfied, in order to significantly reduce the search space.

A constraint solver is an effective procedure that is able to check the satisfiability of constraints and to find solutions which satisfy all of them, if any. Several solvers have been proposed since the 80's, for several constraint domains and with different properties and implementation techniques.

The use of constraints over finite domains allows one to define combinatorial optimization problems, such as scheduling problems, timetabling problems, etc. [19,1]. Constraint solver for such domains are based on the concept of constraint propagation, typically using some version of the arc-consistency technique [17,18], and on the search in the space of all solutions, guided by constraint propagation which allows one to discover partial variable assignments which cannot lead to feasible solutions. For optimization problems, the well-known branch-and-bound technique is used together with constraint propagation to reduce the number of solutions to check before finding an optimal solution.

*Soft constraints and preferences:*

In many real-life problems, not all solutions are equally desirable. Moreover, in some situations the constraints are naturally associated to preferences which tells us how important the constraint is, or how bad it is if we violate it. To faithfully model such situations, the classical constraint formalism has been generalized to allow for the description of preferences. Preferences are just constraints where, for each combination of values to the variables of the constraint, it is associated an element (that is, the preference for that combination) taken from an (partially or totally) ordered set. In the last 10 years various formalisms have been developed to model and handle problems with preferences, among which the most general are the one based on semirings [3,4], the valued CSPs [24], and the so-called CP-nets [5].

Solving a problem with preferences is in general more complex than solving a constraint problem, since now the desired solutions are only those that are optimal according to the preferences. Techniques of branch-and-bound, of constraint propagation adapted to work with preferences rather than classical constraints, and of abstraction, are useful to build a solver for problems with preferences.

One of our goals in this project is to develop new formalisms to model more complex preferences than those handled until now. For example, bipolar preferences [9], which allows for the description of both desires and solutions to be avoided, or also qualitative and quantitative preferences at the same time. We also would like to model and aggregate preferences described by multiple agents.

*Preferences and uncertainty:*

Besides preferences, several real-life problems contain uncertain parts, where the user cannot take decisions but only observe the happening of the events. For this reason, it is important to be able to handle uncertainty in problems with constraints and/or preferences. The goal is then to find one or more solutions which are good according to the preferences but also reasonably compatible with the uncertainty of the problem.

Various formalisms to model uncertainty have been used, depending on the kind of uncertainty considered. For example, one can associate a probability to each event, or, when estimates are not available, a degree of possibility [27,10].

In the literature there are already some approaches to handle fuzzy preferences and uncertainty [21]. We want to extend these approaches and generalize them to any kind of preferences.

*Constraints and scheduling:*

Scheduling problems occur in many real-life situations [2,23,20,15]. In fact, a scheduling problem can be defined as the assignment of start and end times to a set of tasks (or actions), subject to certain constraints. Constraints are typically either time constraints or resource constraints. Coordination of production in a factory, management of space missions, and transportation scheduling to support crisis management are representative examples.

Feasibility alone is seldom the goal of scheduling; in fact usually a set of objectives and preferences have to be optimized. Therefore the scheduling problem is primarily concerned with figuring out when tasks should be executed so that the final solution guarantees good performances.

In this field we are particularly interested in two different aspects: to consider scheduling problems with preferences on the activities, and to obtain solutions which are robust. Regarding the first point we aim at introducing preferences in a state-of-the-art architecture (OOSCAR [6]) as well as in the associated solving methods [7,8]. To produce robust schedules we would like to pursue the idea of building flexible solutions [22], studying the concept of resource flexibility and developing an empirical framework to verify the effectiveness of the combination of robust solutions with the repairs methods used.

*Constraints and multi-objective optimization:*

Multi-objective optimisation is considered important in many real life applications. In combinatorial problems arising from industry or economics, there are typically many aspects that should be optimised. Often, sorting the objective functions in order of importance is simply not possible; moreover objectives can be conflicting and a solution optimal with respect to one objective function could be sub-optimal, or even unacceptably bad with respect to the others.

The problem has been addressed in different research areas: in Artificial Intelligence, in particular in Constraint Programming [13]. In Operations Research [11], using specific solvers (for instance, for scheduling problems [26] [16]), or (integer) linear programming solvers [25], or local search [14], or evolutionary algorithms [12] [28].

Various techniques have been proposed in the past years. A first method is to combine the different objective functions into one, and then use the classic methods for (mono-objective) combinatorial optimisation. All these methods have in common the idea of transforming the partial order among possible solutions given by multi-objective optimisation into a total order. Differences stand in how balanced the proposed solution is with respect to the various criteria, how efficient the method is (for example, how much pruning the method allows for), the possible need to insert a priori normalising factors. These methods are, in general, very efficient, because they need only one optimisation phase, reusing the

same optimisation algorithms used in mono-objective optimisation. Nevertheless, there is no assurance that the found solution is indeed what the user was looking for.

Another class of algorithms provides to the user the whole set of non-dominated solutions, namely, those solutions that cannot be improved with respect to one objective without sacrificing another objective. The set of such solutions is also called the Pareto optimal frontier. Obviously, the Pareto optimal frontier gives to the user the maximal possible information, but it is also less efficient to compute, both in terms of computing time and of memory occupation, with respect to methods that propose only one solution. The advantage is that the user is able to select him/herself a solution, and does not have to accept blindly the solution provided by the optimisation algorithm. In this project we want to study and integrate techniques of Artificial Intelligence and Operation research for multi-objective optimization. The aim is to develop hybrid algorithms which are able to generate efficiently the Pareto optimal frontier.

## **2.4.a Riferimenti bibliografici**

- [1] K. Apt. *Principles of Constraint Programming*. Cambridge University Press, 2003.
- [2] M. Becker and S.F. Smith. *Mixed-Initiative Resource Management: The AMC Barrel Allocator"*. *Proceedings of AIPS-2000*, 2000
- [3] S. Bistarelli, U. Montanari and F. Rossi. *Constraint Solving over Semirings*. *Proc. of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI 95)*, pages 624-630. Morgan Kaufmann, 1995.
- [4] S. Bistarelli, U. Montanari and F. Rossi. *Semiring-based Constraint Solving and Optimization*. *Journal of the ACM*, Vol. 44, No. 2, pages 201-236, 1997.
- [5] C. Boutilier, R. I. Brafman, H. H. Hoos and D. Poole. *Reasoning With Conditional Ceteris Paribus Preference Statements*. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI 1999)*, pages 71-80. Morgan Kaufmann 1999.
- [6] A. Cesta, G. Cortellessa, A. Oddi, N. Policella and A. Susi. *A Constraint-Based Architecture for Flexible Support to Activity Scheduling*. *Lecture Notes in Artificial Intelligence*, N. 2175, 2001.
- [7] A. Cesta, A. Oddi and S.F. Smith. *An Iterative Sampling Procedure for Resource Constrained Project Scheduling with Time Windows*. *Proceedings of IJCAI99*, 1999.
- [8] A. Cesta, A. Oddi and S.F. Smith. *Profile Based Algorithms to Solve Multiple Capacitated Metric Scheduling Problems*. *Proceedings of AIPS98*, 1998.
- [9] D. Dubois, S. Kaci and H. Prade. *Bipolarity in reasoning and decision-An introduction. The case of the possibility theory framework*. *Proc. of Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference (IPMU'04)*, pages 959-966, 2004.
- [10] D. Dubois and H. Prade. *Possibility Theory*. Plenum Press, 1988.
- [11] M. Ehrgott and X. Gandibleux. *A survey and annotated bibliography of multiobjective combinatorial optimization*. *OR Spektrum*, 22:425-460, 2000.
- [12] C.M. Fonseca and P.J. Fleming. *An Overview of Evolutionary Algorithms in Multiobjective Optimization*. *Evolutionary Computation* 3 (1995), no. 1, 1-16.
- [13] M. Gavanelli. *An algorithm for multi-criteria optimization in CSPs*. In Frank van Harmelen, editor, *ECAI 2002. Proceedings of the 15th European Conference on Artificial Intelligence*, pages 136-140, Lyon, France, July 21-26 2002. IOS Press.
- [14] M. Hansen. *Tabu search for multiobjective optimization: MOTS*. *13th International Conference on Multiple Criteria Decision Making (Cape Town)*, January 1997.
- [15] D. Joslin and D. Clements. *Squeaky Wheel Optimization*. *Journal of Artificial Intelligence Research*, vol. 10, 1999.
- [16] C. Le Pape, P. Couronné, D. Vergamini and V. Gosselin. *Time-versus-Capacity Compromises in Project Scheduling*. *Proceedings of the Thirteenth Workshop of the UK Planning Special Interest Group, Strathclyde, United Kingdom*, 1994.
- [17] A.K. Mackworth. *Consistency in Networks of Relations*. *Artificial Intelligence*, Vol. 8, pages 99-118, 1977.
- [18] A. K. Mackworth and E. C. Freuder. *The Complexity of Some Polynomial Network Consistency Algorithms for Constraint Satisfaction Problems*. *Artificial Intelligence*, Vol. 25, No. 1, pages 65-74, 1985.
- [19] K. Marriott, Peter J. Stuckey. *Programming with Constraints: An Introduction*. MIT Press, 1998.
- [20] A. Oddi, N. Policella, A. Cesta, and G. Cortellessa. *Generating High Quality Schedules for a Spacecraft Memory Downlink Problem*. *Proceedings of CP03*, 2003.
- [21] M.S. Pini, F. Rossi and K. B. Venable. *Possibility theory for reasoning about uncertain soft constraints*. *Proc. Eighth European Conferences on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2005)*.



- [22] N. Policella, A. Oddi, S.F. Smith and A. Cesta. *Generating Robust Partial Order Schedules*. Proceedings of CP04, 2004.
- [23] W. Ruml, M.B. Do, and M.P.J. Fromherz. *Online Planning and Scheduling for High Speed Manufacturing*. Proceedings of ICAPS05, 2005.
- [24] T. Schiex, H. Fargier and G. Verfaillie. *Valued Constraint Satisfaction Problems: Hard and Easy Problems*. Proc. of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI 95), pages 631-639. Morgan Kaufmann, 1995.
- [25] R.E. Steuer, *Multiple criteria optimization: Theory, computation, and application*. Wiley, New York, 1986.
- [26] L. N. Van Wassenhove and L. F. Gelders. *Solving a bicriterion scheduling problem*. European Journal of Operational Research 4 (1980), no. 1, 42-48.
- [27] L. A. Zadeh. *Fuzzy Sets as a basis for the theory of possibility*. Fuzzy sets and Systems, pages 13-28, 1978.
- [28] E. Zitzler and L. Thiele. *Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach*. IEEE Transactions on Evolutionary Computation 3 (1999), no. 4, 257-271.

## **2.5 Descrizione del programma e dei compiti dell'Unità di Ricerca**

### **Testo italiano**

*L'attività di ricerca dell'unità di Padova si sviluppa su tre filoni diversi ma altamente correlati tra loro:*

- 1. preferenze e incertezza*
- 2. vincoli e scheduling*
- 3. ottimizzazione multi-obiettivo.*

*Il primo filone si occupa dei formalismi per modellare problemi con preferenze e/o incertezza e delle tecniche per risolvere tali problemi, con lo scopo principale di estendere il potere espressivo sia per le preferenze che per l'incertezza, e di poter gestire anche situazioni dove sia le preferenze che l'incertezza originano da più agenti, studiando le proprietà di funzioni di aggregazione di preferenze.*

*Il secondo filone considera i problemi di scheduling, che sono l'area applicativa di maggior successo per la programmazione con vincoli. Il primo scopo è quello di considerare preferenze sulle attività di un problema di scheduling, il secondo quello di ottenere degli schedule robusti alle modifiche. In entrambi i casi si tratta di gestire preferenze o incertezza all'interno di problemi di scheduling, quindi questo filone è strettamente collegato al primo perché si prevede un uso dei formalismi e delle tecniche sviluppate nel primo filone, adattati ai problemi di scheduling.*

*Nel terzo filone, lo scopo è di capire come risolvere problemi in cui ci sono diversi criteri di ottimizzazione, possibilmente in contrasto tra loro. Questo è molto correlato alla situazione in cui vari agenti esprimono le loro preferenze, e anche alla situazione in cui vari vincoli con preferenze devono essere soddisfatti, entrambe affrontate nel primo filone. Qui però si vuole studiare soprattutto lo sviluppo di algoritmi ibridi che usano tecniche di Intelligenza Artificiale e Ricerca Operativa per produrre la frontiera Pareto ottima. Tali tecniche saranno anche considerate nella prima attività per gestire scenari multi-agente.*

*In tutte le tre linee di ricerca, prima studieremo lo stato dell'arte, poi svilupperemo nuovi formalismi e tecniche, in seguito studieremo le loro proprietà, e infine valideremo alcune linee di lavoro anche sperimentalmente sia su problemi generati casualmente che su piccoli problemi reali.*

*Di seguito descriviamo in dettaglio il programma di lavoro in ognuno dei tre filoni.*

#### *1. Preferenze e incertezza:*

*Le preferenze che si trovano in situazioni reali possono essere di vario tipo: qualitative (come in "Preferisco l'estate all'inverno"), quantitative (come in "Do' voto 8 all'estate e voto 5 all'inverno"), o anche condizionate (come in "Se vivessi in montagna, preferirei l'inverno all'estate"). Inoltre, le preferenze possono essere positive (come in "Preferisco il sole alla pioggia") o anche negative (come in "Non vorrei fare tardi"). Nei problemi reali esistono anche cose non solo preferite, ma obbligatorie, che si esprimono naturalmente tramite vincoli, come in "E' necessario che sia a casa prima delle 8pm".*

*Esistono vari formalismi che sono in grado di descrivere e gestire questi tipi diversi di preferenze e di vincoli. Ad esempio, i vincoli descrivono le cose necessarie, i vincoli soft le preferenze quantitative, le CP-net le preferenze qualitative e condizionate. Inoltre, i formalismi sono anche stati paragonati dal punto di vista del loro potere espressivo, ed è stato visto come approssimare una CP-net con dei vincoli soft. Infine, è stata studiata la coesistenza di preferenze a la CP-net e di vincoli. Ma non esiste nessun formalismo che riesca a gestire tutte queste tipologie di conoscenza allo stesso tempo. Noi vogliamo capire come utilizzare i formalismi esistenti per svilupparne uno che riesca a fare ciò.*

*Questi vari formalismi hanno anche complessità computazionali molto diverse. In particolare, per le CP-net è difficile (NP-hard) controllare se una soluzione è migliore dell'altra, mentre questo è un problema semplice per i vincoli soft. Vogliamo studiare e sfruttare queste differenze per sviluppare algoritmi efficienti per trovare soluzioni ottime e fare test di dominanza.*

Per quanto riguarda le preferenze bipolari, cioè positive e negative, esistono alcuni modi per modellarle, ma noi intendiamo renderli più generali e studiare casi in cui il problema di trovare soluzioni ottime secondo entrambi i tipi di preferenze diventa trattabile. Vogliamo anche evitare di dare priorità ad un tipo di preferenze rispetto all'altro.

Quando le preferenze si uniscono all'incertezza, è necessario trovare soluzioni che siano ottime per le preferenze ma anche robuste rispetto alle parti incerte del problema. Anche se preferenze ed incertezza in alcuni casi possono essere gestite in modo omogeneo tramite la teoria delle possibilità, è opportuno, ai fini di un ordinamento adeguato tra le soluzioni, che tali informazioni vengano tenute separate. Questo permette di riconoscere, per una soluzione, sia il suo livello di preferenza che il suo grado di compatibilità con l'incertezza del problema, che può essere visto come la robustezza della soluzione. Opportuni modi di valutare queste due informazioni possono dare più peso all'uno o all'altro aspetto di una soluzione.

È molto comune che più agenti vogliano esprimere le loro preferenze su alcuni oggetti condivisi. In questi casi, è necessario aggregare le loro preferenze per trovare le soluzioni che possano soddisfare tutti al meglio. Varie funzioni di aggregazione possono essere definite, con proprietà molto diverse tra loro, sia in termini di complessità computazionale che in termini di espressività e adeguatezza. Ad esempio, proprietà interessanti di aggregazioni di preferenze, che sono già state studiate in contesti sociali, sono la fairness e la non-manipolabilità. Vogliamo studiare queste ad altre proprietà e capire anche i loro risvolti sulla complessità della funzione di aggregazione. In altri termini, ci interessa capire quanto dobbiamo pagare in termini di complessità per poter avere una funzione di aggregazione di preferenze che abbia una di queste proprietà desiderate. È anche interessante capire se i classici teoremi di impossibilità che riguardano queste proprietà nel campo sociale si trovino anche nelle situazioni da noi considerate.

## 2. Vincoli e scheduling:

Una soluzione ad un problema di scheduling deve essere eseguita così come prevista per mantenere le qualità della soluzione. Questo è però reso impossibile dalla necessità di eseguire la soluzione in ambienti dinamici dove eventi imprevisti possono velocemente invalidare la soluzione iniziale. Quindi la vita di uno schedule tende ad essere molto corta, così come le qualità ad essa associate. Per ovviare a questo problema si rende necessario studiare soluzioni cosiddette robuste che sono capaci di assorbire, se non tutti, almeno parte dei possibili eventi che si possono verificare durante l'esecuzione.

Tre aspetti si possono distinguere nel concetto di robustezza: (1) la capacità di assorbire eventi imprevisti, (2) la capacità di produrre rapidamente una nuova soluzione quando necessario, (3) la capacità di preservare la qualità delle soluzioni. I primi due aspetti possono essere soddisfatti adottando soluzioni temporalmente flessibili, mentre il terzo può essere raggiunto utilizzando metodi specifici per la generazione di soluzioni flessibili di buona qualità. Su questa base, si vuole analizzare il concetto di flessibilità sulle risorse in modo da estendere la struttura a vincoli che sottosta alle soluzioni temporalmente flessibili. In aggiunta, si vuole sviluppare un apparato sperimentale per validare la combinazione tra soluzioni flessibili e i metodi di riparazione.

Negli esempi presenti in letteratura i problemi di scheduling considerano vari aspetti globali della soluzione quali il makespan (il tempo di completamento di tutte le attività). Un aspetto interessante da considerare sono problemi di scheduling con differenti livelli di preferenza per ogni attività. Questo tipo di problemi ha grande riscontro nel mondo reale. Ad esempio nella gestione di un satellite di comunicazione, servire una particolare richiesta può essere più redditizia di altre o dovuta in seguito a diversi valori di priorità.

Considerare attività con diverse preferenze in problemi di scheduling richiede di introdurre nuovi metodi risolutivi. In particolare, il nostro principale obiettivo è analizzare la complessità dell'integrazione di tecniche attuali basate su attività con diverse preferenze in architetture di scheduling. Ad esempio, un modo per la risoluzione di problemi di scheduling si basa su metodi iterativi in cui soluzioni parziali vengono passo passo raffinate rimuovendo i vari conflitti individuati. L'analisi delle preferenze può essere integrata a questo livello con gli attuali metodi (ad esempio, profilo di risorse). Un approccio alternativo invece si basa sull'utilizzo della flessibilità temporale delle soluzioni. Infatti, è possibile sfruttare l'insieme di allocazioni per ogni attività che soluzioni flessibili assicurano, per considerare, in una fase separata, il soddisfacimento delle preferenze.

## 3. Ottimizzazione multi-obiettivo:

In questo progetto, si desiderano studiare ed integrare tecniche di Intelligenza Artificiale e di Ricerca Operativa per l'ottimizzazione multi-obiettivo. Lo scopo è quello di sviluppare algoritmi ibridi in grado di generare efficientemente la frontiera Pareto ottima.

Preferenze ed ottimizzazione multi-obiettivo possono essere viste come due aspetti di uno stesso concetto: ad assegnamenti (eventualmente, parziali), vengono associati livelli di soddisfazione o valori di una funzione obiettivo. Mentre nell'ottimizzazione multi-obiettivo il valore di ogni funzione è tipicamente un numero reale, nelle preferenze può essere un valore preso da qualunque insieme ordinato (eventualmente, con ordinamento parziale). Storicamente, le preferenze sono state studiate di più nell'Intelligenza Artificiale, mentre l'ottimizzazione multi-obiettivo è tradizionalmente approfondita in Ricerca Operativa.

Spesso gli approcci più efficienti per risolvere problemi di ottimizzazione combinatoria sono approcci ibridi, che nascono dalla integrazione di algoritmi sviluppati nella Ricerca Operativa con quelli tipici dell'Intelligenza Artificiale. Tipicamente, gli algoritmi sviluppati in Programmazione a Vincoli sono particolarmente orientati alla soddisfacibilità del problema, cioè a trovare una soluzione che soddisfa tutti i vincoli. Gli algoritmi sviluppati in Ricerca Operativa sono particolarmente efficienti nell'ottimizzazione: utilizzano all'interno della ricerca branch-and-bound rilassamenti del problema, decomposizioni, cutting planes. I due tipi di approcci sono complementari da questo punto di vista ed algoritmi ibridi che cercano di sfruttare gli aspetti migliori di entrambi i campi sono risultati particolarmente efficienti e sono riusciti ad ottenere soluzioni che nessuna delle due tecniche, prese separatamente, era riuscita a produrre in tempi ragionevoli.

Altri approcci utilizzati in Ricerca Operativa utilizzano algoritmi di ricerca locale, che hanno dimostrato di essere molto veloci nell'ottimizzazione, anche se sono tipicamente incompleti. Anche per questi algoritmi sono state studiate integrazioni con la

programmazione a vincoli. Particolarmente interessante, ai fini del progetto, è il metodo chiamato *large neighbourhood search*, che esplora in maniera esaustiva l'intorno della configurazione corrente utilizzando la programmazione a vincoli. Usando la programmazione a vincoli per esplorare l'intorno è possibile utilizzare intorni di dimensione più grande ed esplorarli all'ottimo, il che permette di arrivare a soluzioni migliori prima di essere bloccati in minimi (o massimi) locali.

La ricerca proposta è divisa in quattro fasi.

In una prima fase, si studieranno gli algoritmi di ottimizzazione multi-obiettivo proposti in letteratura nelle diverse aree dell'intelligenza artificiale e della ricerca operativa. Si porrà particolare attenzione ai problemi di ottimizzazione combinatoria, così come sono sviluppati nelle due aree. Si studieranno in particolare gli algoritmi proposti nella programmazione a vincoli, nella programmazione lineare intera, nella ricerca locale e nei metodi basati su popolazione (in particolare, algoritmi genetici).

Nella seconda fase, si studieranno le tecniche sviluppate fino a questo momento per integrare algoritmi di ricerca operativa (in particolare, ricerca locale e programmazione lineare intera) con programmazione a vincoli al fine di ottenere soluzioni in tempi più brevi o di ottenere soluzioni migliori a parità di tempo di calcolo. In particolare, le tecniche che appaiono più promettenti sono le tecniche di decomposizione di problemi in sottoproblemi (ad esempio, decomposizione di Benders) e ricerca locale con esplorazione di uno spazio ampio delle configurazioni vicine (*large neighborhood search*).

Nella terza fase, si studierà l'applicabilità delle tecniche di integrazione fra ricerca locale e programmazione lineare intera con programmazione a vincoli studiate nella seconda fase, all'ottimizzazione multi-obiettivo. In particolare, verranno integrati algoritmi di ottimizzazione multi-obiettivo di programmazione a vincoli con algoritmi di ottimizzazione mono e multi-obiettivo delle altre aree. Lo scopo di questa terza fase è quello di ottenere algoritmi in grado di generare l'intera frontiera Pareto-ottima o una sua caratterizzazione accettabile in tempi brevi. Tale caratterizzazione potrebbe essere un sottoinsieme significativo dei punti che costituiscono la frontiera, oppure insiemi di punti che, benché non appartenenti alla frontiera, ne siano sufficientemente vicini.

All'interno di tutte queste fasi, si terrà in particolare considerazione la rappresentazione dei dati in memoria. Specificamente, la frontiera Pareto ottima potrebbe contenere un numero molto elevato di soluzioni. Si dovranno utilizzare strutture dati appropriate che permettano di accedere velocemente ai dati in memoria, utilizzando indicizzazioni hash o con alberi di ricerca.

La quarta fase consiste nella validazione degli algoritmi sviluppati, che verrà fatta attraverso sperimentazione, sia utilizzando i benchmark standard disponibili, sia utilizzando problemi reali, in particolare di scheduling e/o di pianificazione di orario.

#### **Testo inglese**

The research activity of the Padova unit will develop around three different lines of work, which are however very correlated among them:

1. preferences and uncertainty
2. constraints and scheduling
3. multi-objective combinatorial optimization.

The first line will focus on formalisms to model problems with preferences and/or uncertainty and on techniques to solve such problems, with the main aim to extend the expressive power both for preferences and for uncertainty, and to be able to handle also situations where both preferences and uncertainty are generated by several agents, while studying properties of preference aggregation functions.

The second line of work will consider scheduling problems, which are the main application area for constraint programming. The first goal is to consider preferences over the activities in a scheduling problem, the second goal is to obtain schedules which are robust with respect to changes. In both cases, we need to handle preferences and uncertainty within scheduling problems, thus this line of work is tightly related to the first one, since we envision the use of the formalisms and the techniques developed in the first line, suitably adapted to work for scheduling problems.

In the third line of work, the goal is to understand how to solve problems which have several different optimization criteria, possibly contradicting each other. This is very related to the situation where several agents express their preferences, and also to the scenario where several constraints with preferences need to be satisfied, both considered in the first line of work. Here, however, we want to study mainly the development of hybrid algorithms which use both Artificial Intelligence and Operation Research techniques to generate the Pareto optimal frontier. Such techniques will also be considered in the first line of work to handle multi-agent scenarios.

In all the three lines of research, we will first survey the state of the art. Then we will develop new formalisms and techniques, and we will study properties of our developments. Finally we will validate some of the work also experimentally on both randomly generated problems and on small real-life problems.

In the following we will describe the activities we plan to do in each of the three lines of work.

1. Preferences and uncertainty

The kind of preferences that occur in real-life problems may be very different: they can be qualitative, as in "I prefer summer to winter", quantitative, as in "I prefer summer at level 8 and winter at level 5", or also conditional, as in "If I lived in the mountains, I would prefer winter to summer". Moreover, preferences can be positive, as in "I prefer a sunny day to a rainy one", or also negative, as in "I would not like to be late". Finally, in real-life problems there are also

scenarios which are not only preferred, but customary, and that are naturally modelled via constraints, as in "I need to be home before 8pm".

There are several formalisms which are able to describe each one of these different kinds of preferences. For example, constraints can describe the necessary scenarios, soft constraints can model quantitative preferences, CP-nets can express qualitative and conditional preferences. Moreover, these formalisms have been compared from the expressive point of view, and it has been seen how to approximate a CP-net with soft constraints. Finally, the co-existence of preferences a la CP-net and a la constraints has been studied. However, there is no formalism that can handle all these different kinds of knowledge at the same time. We want to understand how to exploit the existing frameworks to develop a new one that can do this.

The existing formalisms are very different also for their computational complexity. In particular, for CP-nets it is difficult (NP-hard) to check whether one solution is better than another one, while this is an easy problem for soft constraints. We want to study and to exploit these differences to develop efficient algorithms to find optimal solutions and to perform dominance tests.

As for the bipolar preferences, that is, preferences that can be both positive and negative, there are some ways to model them, but we are interested in generalizing them and in studying cases in which the problem of finding optimal solutions according to both types of preferences becomes tractable. We also want to avoid to give priority to one type of preferences with respect to the other one.

When preferences and uncertainty coexist, it is necessary to find solutions which are optimal for the preferences but also robust with respect to the uncertain parts of the problem. Even if preferences and uncertainty in some cases can be handled homogeneously via possibility theory, it is useful, to achieve a better ordering among the solutions, that such information is kept separate. This allows to recognize, for a solution, both its level of preference and its degree of compatibility with the uncertainty of the problem, which can be seen as the robustness of the solution. Suitable methods to evaluate these two kinds of information can give priority to one of the other aspect of a solution.

It is very common that several agents need to express preferences on some objects. In these cases, it is necessary to aggregate their preferences to find solutions that can satisfy all of them. Several aggregation functions can be defined, with very different properties, both in terms of computational complexity and in terms of expressivity and suitability. For example, interesting properties of aggregation functions, which have been already studied in social contexts, are fairness and non-manipulability. We would like to study these and other properties and understand also their consequences on the computational complexity of the aggregation function. In other words, we want to understand how much we need to pay in terms of computational complexity to have an aggregation function that has one of these desirable properties. It is also interesting to understand if the classical impossibility theorems which concern such properties in the social sciences hold also in the scenarios considered by our study.

## 2. Constraints and scheduling

Though a schedule offers the potential for a more optimized execution than would otherwise be obtained, it must in fact be executed as planned to achieve this potential. In practice this is generally made difficult by a dynamic execution environment, where unforeseen events quickly invalidate the schedule's predictive assumptions and bring into question the continuing validity of the schedule's prescribed actions. The lifetime of a schedule tends to be very short, and hence its optimizing advantages are generally not realized.

Robustness entails three different aspects: (1) the ability of absorbing unexpected events, (2) the ability of rapidly producing a new solution when necessary, (3) the ability of preserving quality solutions. The first two aspects can be satisfied by using time-flexible solutions while the third one can be obtained using specific solving process to achieve good quality time-flexible solutions. On this basis we would like analyze the concept of resource flexibility to extend the constraint based structure underlying the time-flexible solutions. Additionally, we will work on developing an empirical framework that is necessary to verify the effectiveness of the combination of flexible solutions with the repairs methods used.

Approaches to scheduling problem mostly consider global aspects of the solution like makespan, tardiness, and so on. Novel approaches are required to take into account scheduling problem with different levels of preference for each activity (based for instance on different activity rewards). This kind of problem are often present in real world application. For instance, a first example is given by the oversubscribed scheduling problem (e.g., satellite slot allocation) where it is necessary to make a decision on which activity to dump. Preferences require then solving strategies able to distinguish the additional variable that they introduce.

The goal of considering activity preferences into scheduling problem can be faced at the first stage by introducing novel solving methods. In particular, our aim is to analyze the complexity of integrating current techniques based on activity preferences in scheduling frameworks. For example, a current way to solve scheduling problem is based on iterative methods where at each step a partial solution is analyzed to discover possible flaws and then to solve them. The analysis of preferences satisfaction can be introduced at this level together with the current methods (i.e., resource profiles). A different approach instead may exploit the production of time-flexible solutions. In fact, it is possible to use the set of allocations for each activity that these solution ensure, to consider, as a separated phase, the fulfillment of the preferences.

## 3. Multi-objective optimization

In this project, we will study and integrate algorithms and techniques of Artificial Intelligence and Operations Research for multi-objective optimisation. The aim is to develop hybrid algorithms able to efficiently generate the Pareto optimal frontier.

Preferences and multi-objective optimization can be seen as two facets of a same concept: in both cases (partial) assignments are associated to satisfaction levels or values of an objective function. While in multi-objective optimization the value is typically a real number, in preferences it can be a value taken from any partially ordered set. Traditionally, preferences are more studied in Artificial Intelligence, while multi-criteria optimization is studied in Operations Research. There are also notable examples of

integration of the two concepts.

Often, the most efficient algorithms for combinatorial optimisation are hybrid, and integrate Artificial Intelligence and Operations Research techniques. Typically, algorithms in Constraint Programming are very biased toward satisfiability, that is finding a solution satisfying all constraints. Algorithms developed in Operations Research are very efficient in optimisation: they use relaxations of the problem, decompositions, or cutting planes in the branch-and-bound search. The two approaches are complementary from this viewpoint. Hybrid algorithms exploiting the features of both approaches have proved very efficient, and discovered solutions that neither of the two techniques, taken separately, was able to provide in reasonable time.

Other approaches of Operations Research are based on local search algorithms, that are very fast in optimisation, although they are typically incomplete. Local Search algorithms have also been integrated with constraint programming. In particular, Large Neighbourhood Search, that explores exhaustively a neighbourhood with constraint programming, is very interesting for the purposes of the project. By using constraint programming to explore the neighbourhood, larger neighbourhoods can be used and explored to optimality, which leads to better solutions before getting stuck in local minima (or maxima).

The research is divided into four phases.

In the first phase, multi-objective algorithms in the literature of Artificial Intelligence and Operations Research will be studied. Most attention will be given to algorithm for combinatorial optimisation, as they are addressed in the two research areas. In particular, we will study algorithms developed in the areas of constraint programming, integer linear programming, local search, and population based methods (such as genetic algorithms).

In the second phase, we will study techniques for integrating Operations Research algorithms (in particular, local search and integer linear programming) with constraint programming developed for mono-objective optimisation in order to obtain either solutions in shorter time or better solutions. In particular, very promising are decomposition techniques (like Bender's decomposition) and local search with large neighborhood.

In the third phase, we will study applicability of the integration techniques studied in the second phase to multi-objective optimisation. In particular, we will integrate algorithms of multi-objective optimisation developed in Constraint Programming with algorithms of mono and multi-objective optimisation of local search and integer linear programming. The aim of this third phase is to obtain algorithms able to generate the whole Pareto optimal frontier or an acceptable characterisation in reasonable time. Such a characterisation could be either a significant subset of the points in the frontier, or a set of solutions that are close enough to the frontier for the considered application.

In all of these phases, we will consider as an important issue the data structures. In fact, the Pareto optimal frontier might contain a huge number of solutions. Thus, efficient data structures might be necessary in order to access the data; we are thinking in particular of hash indexing or special purpose trees.

The fourth phase is validation of the developed algorithms. It will be carried out through experiments, using both publicly available benchmarks, and real-life problems, like scheduling and/or timetabling problems.

---

## **2.6 Descrizione delle attrezzature già disponibili ed utilizzabili per la ricerca proposta con valore patrimoniale superiore a 25.000 Euro**

### **Testo italiano**

Nessuna

### **Testo inglese**

Nessuna

## **2.7 Descrizione delle Grandi attrezzature da acquisire (GA)**

### **Testo italiano**

Nessuna

### **Testo inglese**

Nessuna

## 2.8 Mesi uomo complessivi dedicati al programma

### Testo italiano

		Numero	Mesi uomo 1° anno	Mesi uomo 2° anno	Totale mesi uomo
<i>Personale universitario dell'Università sede dell'Unità di Ricerca</i>		2	13	13	26
<i>Personale universitario di altre Università</i>		1	4	4	8
<i>Titolari di assegni di ricerca</i>		0			
<i>Titolari di borse</i>	<i>Dottorato</i>	1	6	6	12
	<i>Post-dottorato</i>	0			
	<i>Scuola di Specializzazione</i>	0			
<i>Personale a contratto</i>	<i>Assegnisti</i>	0			
	<i>Borsisti</i>	1	4	4	8
	<i>Dottorandi</i>	0			
	<i>Altre tipologie</i>	0			
<i>Personale extrauniversitario</i>		3	7	7	14
<b>TOTALE</b>		<b>8</b>	<b>34</b>	<b>34</b>	<b>68</b>

### Testo inglese

		Numero	Mesi uomo 1° anno	Mesi uomo 2° anno	Totale mesi uomo
<i>University Personnel</i>		2	13	13	26
<i>Other University Personnel</i>		1	4	4	8
<i>Work contract (research grants, free lance contracts)</i>		0			
<i>PHD Fellows &amp; PHD Students</i>	<i>PHD Students</i>	1	6	6	12
	<i>Post-Doctoral Fellows</i>	0			
	<i>Specialization School</i>	0			
<i>Personnel to be hired</i>	<i>Work contract (research grants, free lance contracts)</i>	0			
	<i>PHD Fellows &amp; PHD Students</i>	1	4	4	8
	<i>PHD Students</i>	0			
	<i>Other tipologies</i>	0			
<i>No cost Non University Personnel</i>		3	7	7	14
<b>TOTALE</b>		<b>8</b>	<b>34</b>	<b>34</b>	<b>68</b>

### PARTE III

#### 3.1 Costo complessivo del Programma dell'Unità di Ricerca

##### Testo italiano

Voce di spesa	Spesa in Euro	Descrizione
Materiale inventariabile	4.500	Due PC piu' due portatili. Si assume un costo di 1000 euro per ogni PC e 1250 euro per ogni portatile.
Grandi Attrezzature		
Materiale di consumo e funzionamento	4.200	Spese telefoniche, carta, toner.
Spese per calcolo ed elaborazione dati		
Personale a contratto	9.000	Quattro mesi uomo l'anno per una borsa di studio per sviluppo ed implementazione di risolutori per vincoli e preferenze.
Servizi esterni		
Missioni	25.000	Missioni all'estero per partecipazioni a convegni, o visite a collaboratori di lavoro. Stima di circa 1500 euro a missione.
Pubblicazioni	0	
Partecipazione / Organizzazione convegni	9.000	Iscrizioni a conferenze e spese per la disseminazione dei risultati del lavoro. Stima di circa 500 euro per ogni iscrizione a conferenza.
Altro		
<b>TOTALE</b>	<b>51.700</b>	

##### Testo inglese

Voce di spesa	Spesa in Euro	Descrizione
Materiale inventariabile	4.500	Two PCs plus two portable PCs. We assume a cost of 1000 euros for each PC and 1250 euros for each portable PC.
Grandi Attrezzature		
Materiale di consumo e funzionamento	4.200	Telephone, paper, toner.
Spese per calcolo ed elaborazione dati		
Personale a contratto	9.000	Four man-months each year for a scholarship to formalize and develop solvers for constraints and preferences.
Servizi esterni		
Missioni	25.000	Trips abroad for participation to international conferences, or visits to collaborators. Estimate of about 1500 euros for each trip.
Pubblicazioni	0	
Partecipazione / Organizzazione convegni	9.000	Conference fees and expenses related to conference participation and dissemination of the results of the project. Estimate of 500 euros for each conference fee.
Altro		
<b>TOTALE</b>	<b>51.700</b>	

#### 3.2 Costo complessivo del Programma di Ricerca

		Descrizione
Costo complessivo del Programma dell'Unità di Ricerca	51.700	
Fondi disponibili (RD + RA) comprensivi dell'8% max per spese di gestione	15.600	RA: 15600 euro, di cui 4200 come 8% per spese di gestione.
Cofinanziamento di altre amministrazioni		

<b>Cofinanziamento richiesto al MIUR</b>	36.100
--	--------

**3.3.1 Certifico la dichiarata disponibilità e l'utilizzabilità dei fondi di Ateneo (RD e RA)**

SI

*(per la copia da depositare presso l'Ateneo e per l'assenso alla diffusione via Internet delle informazioni riguardanti i programmi finanziati e la loro elaborazione necessaria alle valutazioni; D. Lgs, 196 del 30.6.2003 sulla "Tutela dei dati personali")*

Firma \_\_\_\_\_

Data (dal sistema alla chiusura della domanda)