Plenary Sessions Abstracts

Thursday May 29,2008

Plenary Session T1

Assignment problems: The first fifty years

Silvano Martello

In 1955. H.W.Huhn published the Hungarian algorithm, the first polynomial-time algorithm for the assignment problem. In the next fifty years assignment problems attracted hundreds of researchers. Their studies accompanied and sometimes anticipated the development of Combinatorial Optimization, producing fundamental contributions to all algorithmic techniques in use nowadays. This talk is based on the forthcoming book by R. Burkard, M. Dell'Amico and S. Martello., *Assignment Problems* (SIAM). We review the most important assignment problems (linear, quadratic, bottleneck), and the most relevant results in this area. Book home page: http://www.assignmentproblems.com

Plenary Session T2

A history tour on the exact solution of the Quadratic Assignment Problem

Catherine Roucairol

The QAP (Quadratic Assignment Problem) consists of assigning *n* facilities to *n* locations in order to minimize the total weighted cost of interaction between facilities. It is well known that it is a very difficult Combinatorial Optimization problem to solve exactly. Only some instances of size around n=30 are solved to optimality.

The main difficulty is due to the lack of a sharp and efficient lower bound technique in the Branch and Bound method. A good trade-off between the quality of the lower bound and the time to perform it is hard to find. We will review and discuss several lower bounds both in term of quality and computational time needed: linearization techniques, Convex Quadratic Programming, Semi Definite Programming...

We will also point out some recent developments like branching strategies which have resulted in an improvement in the ability to solve QAPs exactly.

As the number of nodes in the Branch and Bound (B&B) tree becomes huge (millions of nodes for instance of size n greater than 25), exact solutions methods have been implemented on high-performance computers. The idea is that if progresses have to be done on the B&B method in itself, the use of new powerful parallel systems like clusters and grid of machines could help in between.

We will survey the strategies of parallelization used and the best computational results obtained for the problem.

Friday May 30, 2008

Plenary Session F1

Developing and Testing General-purpose Combinatorial Optimization Platforms Based on Metaheuristic Methodology

Manuel Laguna

The success of applying metaheuristics to combinatorial optimization has been well-documented. From scheduling to vehicle routing and data mining applications, the development of context-specific metaheuristics has led to numerous publications in the last twenty years. Although the original framework of some of theses metaheuristics — for instance genetic algorithms — was based on a context-independent design, the general research trend for a number of years has been the implementation of procedures that are highly specialized for increasingly narrower classes of combinatorial problems. The notable exception to this trend occurs in the world of commercial software, where metaheuristics have become the solution engines for a few general-purpose optimizers (e.g., Evolver from Palisade Corporation and OptQuest from OptTek Systems). We discuss our experiences with the development of general-purpose metaheuristic procedures for combinatorial optimization, for both academic and commercial purposes. The solution to a fair number of combinatorial optimization problems may be represented as a permutation of elements or as a binary string. We focus on these two solution representations and describe the design process that we followed to create procedures based on metaheuristic methodology for approaching these fairly general problem classes. The work is based on collaboration with Abraham Duarte (Universidad Rey Juan Carlos), Francisco Gortazar (Universidad Rey Juan Carlos), and Rafael Martí (Universidad de Valencia).

Plenary Session F2

On Cost Allocation Schemes in Networks: Cooperative Game Theory and Combinatorial Models

Darko Skorin-Kapov

New technologies prompted an explosion in the development of networks. The modern network design techniques often lead to a construction of the most profitable or the least cost network that provides some service to customers. Some examples of classes of such networks include multicasting networks in telecommunications, hub networks in air transportation, road delivery networks, and oil and gas pipeline networks. There are various costs and gains associated with building and using such service networks and the involved multiple network users and/or owners possibly have conflicting objectives. However, they might cooperate in order to decrease their joint cost or increase their joint profit. These individuals or organizations will support a globally ’attractive’ solution(s) only if their expectations for a ’fair share’ of the cost or profit are met. Providing network developers, users and owners with efficiently computable ’fair’ cost (profit) allocation solution procedures is of great importance for strategic network management.

A common approach in the literature is the formulation of the associated network cost allocation problem as a cooperative game in characteristic function form, followed by the evaluation of various solution concepts such as core, nucleolus, the least -core, Shapley value, etc. These game theoretic solution concepts are computationally prohibitive even for relatively small problems and there are no general practical algorithms for the computation of these solutions. Consequently, researchers have concentrated on individual classes of games to demonstrate that computation of cost allocation solution concepts is sometimes feasible in the context of a particular problem.

We overview some previously published, as well as some new results, in the development of algorithmic cooperative game theory based mechanisms to efficiently compute 'attractive' cost allocation solutions for various classes of service networks. We also present some open cost allocation problems that have emerged out of recent networks developments and the potential contribution that combinatorial game theory models might make in the future.

Saturday May 31, 2008

Plenary Session S1

Polynomial Time Approximation Schemes for Bicriteria Optimization Problems: A New Direction for Integer Programming

Asaf Levin

Goemans and Ravi (SWAT 96) considered the constrained minimum spanning tree problem, in which we are given an undirected graph G = (V,E) where each edge is associated with a length and a cost. We are also given a length bound L. The goal is to find a spanning tree whose total length is at most L, and its cost is minimized. This is an example of the bicriteria optimization problems that we consider. We have an underlying easy optimization problem (e.g. minimum spanning tree) and one complicated constraint (that can be seen as a bound on one objective), while we try to optimize another objective function. This complicated constraint makes the problem NP-hard, and hence we devise polynomial time approximation schemes (PTAS). Following Goemans and Ravi the PTAS is based on Lagrangian relaxation with some initial guessing step. I will present algorithms for various problems that follow the framework of Goemans and Ravi and extend it to allow smart patching methods for other problems. The patching methods motivate the study of the edge directions of the underlying easy polytopes, and I will discuss these connections.

Plenary Session S2

Submodular Functions, Matroids, and Certain Polyhedra

Jack Edmonds

A tutorial on the optimization theory of matroids and submodular set functions. Related expositions are in the books, "Combinatorial Optimization", respectively by Schriver; by Korte and Vygen; and edited by Junger, Reinelt, and Rinaldi. (Little on the subject is to be found in works on matroids per se since the subject is regarded in math circles as mainly axiomatics concerned with linear and algebraic dependence.)

Parallel Sessions Abstracts

Note: The presenting author is indicated with *

Thursday May 29,2008

T1A.1

Car sequencing problem in the robotic assembly line

Grzegorz Pawlak, Jacek Blazewicz, Slawomir Bak*

The modern car factory generally consists of three main production shops. The body shop where the body of the car is built, paint shop where the car body is painted and the assembly line, where the car is assembled, mostly manually. Our paper is considering the automatic body assembly line in the body shop stage. This line consist of sequence of the robotic stages where body is constructed. In this line the montage is purely automatic. Mostly, there are welding and spot welding operations. The robot stages are flexible in the sense of servicing many different kinds of car bodies. The processing time differs for the stage and for the body type. The main criteria for the car factory is the throughput rate maximization. This criterion depends on the car sequence entering the line. Generally, the car sequence is constructed in the production planning phase. Because there is no buffer inside the automatic assembly line planned sequence could not be changed during the production process at that stage.

The car sequencing problem was widely considered in the literature. The problem was formulated and described in different ways depends on the stage which was analyzed in the car producing factory. Practical car sequencing problem was considered and taken from the real car plant. The goal was to maximize the total completion time in the body shop, specifically for the robotic assembly line in this shop. From the scheduling point of view it is total completion time criterion for the car sequence.

From the complexity point of view the general car sequencing problem is NP-hard in the strong sense. The multi stage flow shop is also NP-hard in the strong sense. In the paper the special case flow shop problem is considered where the tasks processing times are restricted. In the paper the Branch and Bound method and some heuristics algorithms were proposed. Then the known NEH heuristics was used as an initial solution for the Local Search and Tabu Search methods. These algorithms were compared in the computational experiment due to their effectiveness.

T1A.2

Scheduling workflow applications on a grid

Marek Mika, Grzegorz Waligóra, and Jan Węglarz*

We consider a problem of scheduling workflow applications on a grid. The objective of our research is to formulate a model of the problem based on a project scheduling oriented approach, and to propose a method of approaching the problem as an optimization problem. Although the model we develop is general and allows the scheduling of any set of tasks on any set of resources, we consider so-called workflow applications because of their particular practical importance. Workflow applications can be viewed as complex sets of precedence-related various transformations (tasks) performed on some data. They are mostly scientific, data intensive applications which, because of large amounts of computations and data involved, require high computing power to be executed efficiently. Although different criteria may be considered to evaluate schedules, we focus on a time criterion, more precisely, we try to minimize the time of execution of a given workflow (i.e. the completion time of a given set of tasks).

The problem of scheduling workflow applications across many sites on a grid is very complex, especially when the network capacity varies between the sites. In addition, we often do not possess complete information about the jobs. The process of obtaining a performance model of a job is not trivial. In particular, the processing times of all tasks on different computer systems (grid resources) are not easy to evaluate. Also other parameters (e.g. bandwidth, resource availability, etc.) may change quite rapidly in grid environments. Thus, generally, we deal with scheduling under uncertainty. In this work we show under what simplifying assumptions it is possible to bring the considered problem to a deterministic scheduling problem. More precisely, under these assumptions we formulate a model of the problem based on classical project scheduling models. It is easy to see the similarity between a workflow and a project. In both cases we have a set of precedence-related tasks (activities) which are to be scheduled on a given set of resources. Thus, it is justified to describe the problem of scheduling workflows on a grid in terms of project scheduling.

In the previous works we assumed that we had only one workflow to schedule, and that network was not overloaded, i.e. tasks did not have to compete for network resources. In consequence, we treated transmission time simply as setup times, and network was not a scarce resource. In this work we modify the model towards a more realistic one. To this end, we consider network as a resource for which tasks have to apply. As a result, we distinguish transmission tasks as a separate type of tasks, which can compete for a given network connection. More precisely, bandwidth is a network resource which can be divided among many transmission tasks. Thus, we present a more practical model of the problem, where there are two type of tasks: computational tasks, competing for computational resources (processors), and transmission tasks, competing for network resources of a grid.

For the presented model of the problem of scheduling workflow applications on a grid, we formulate an approach to searching for its optimal solution. The approach consists of three phases: (1) to find feasible assignments of grid computational resources to computational tasks of a workflow, with respect to their processor requirements, (2) to examine which of the assignments are feasible with respect to grid network resources, i.e. the connections between grid computational nodes have to meet the network requirements of transmission tasks of the workflow, and (iii) to find a feasible solution with respect to phases (1) and (2) minimizing the total execution time of the workflow. We propose a general approach to attack the defined problem, and some examples justifying its practical importance. The next step ahead is constructing efficient heuristic algorithms for solving the defined problem.

T1A.3

Implicit Cooperation in Multi-Organization Clusters

Daniel Cordeiro[*], Denis Trystram, Alfredo Goldman

Commodity hardware and the maturity of grid middleware systems available today allow the development of large distributed systems composed by many parallel jobs. In order to fully utilize the available hardware and get the best performance, we need a sophisticated scheduling model that can respect the cluster owner's best interest and, at the same time, achieve global performance goals.

Grid computing systems are composed by organizations that own clusters of computers. A grid user submit his/her jobs to a scheduler system that can choose any machine available in any of these clusters. However, each organization (a university lab, for instance) that shares its resources wants to take maximum advantage of its own hardware. In order to improve cooperation between organizations, local jobs must be prioritized.

The scheduling of the jobs in the available grid machines is a crucial problem. Although each user submits jobs locally in his/her own organization, it is necessary to optimize the allocation of the jobs for the whole platform in order to achieve good performance.

Pascual et al. proposed a preliminary analysis of the scheduling problem of rigid parallel tasks in multi-organization, homogeneous clusters. In their work, they proposed a new algorithm that, using an initial scheduling using Highest First order done locally by each organization, produces a global schedule that does not delay the local makespan of any organization. Their main contribution was an algorithm for this centralized global controller that can always be profitable. They showed that their algorithm is a 3-approximation which is asymptotically the best that can be obtained for this specific problem.

In this work, we extend the results obtained by Pascual et al. to organizations that run sequential jobs. Starting from the heuristic proposed by Pascual, we study the effects of allowing each organization to choose its own scheduling objective (such as total completion time and sum of completion times) in the global scheduling.

T1A.4

Timetabling and discrete tomography

Dominique de Werra

Starting from the basic class-teacher timetabling problem with unavailability constraints we shall show how it can be generalized in order to get a model able to apprehend the famous image reconstruction problem occurring in discrete tomography. It consists essentially in reconstructing an image given by its projections (say vertical and horizontal) , i.e., the numbers of pixels of each colour occurring in each line (row or column) of the rectangular array associated to the image.

Some additional requirements will be introduced and the complexity status of the resulting problems will be studied. In addition we will consider some applications to a few specific scheduling problems.

T1B.1

Some structural properties of a least central subtree of a tree

Martti Hamina[*] and Matti Peltola

The "middle part" of a graph has important applications in transportations (facility planning and location problems). Much research has been devoted to define that "middle part" of a tree. The solutions are usually limited to special types of "middle part" of a tree, like central points or central paths.

We consider centrality concept, the subtree center of a tree. The concept does not restrict the structure of the "middle part" of a tree. It can be a point or a path or some other kind of subtree such that the subtree is the most central when compared with all subtrees of the tree.

For every tree T there is a joinsemilattice L(T) of subtrees of T, where the meet of subtrees S_1 and S_2 equals the subtree induced by the intersection of the point sets of S_1 and S_2 whenever the intersection is nonempty and the join of subtrees S_1 and S_2 is the least subtree of *T* containing the subtrees S_1 and S_2. Note that the empty graph is not a subtree of *T*, and thus, in general, there is no least element in *L(T)*. The distance in the joinsemilattice *L(T)* is the same as the distance in the (undirected) Hasse diagram graph G_L of *L(T)*.

A subtree *S* of a tree *T* is the central subtree of *T*, if *S* has the minimum eccentricity in the joinsemilattice *L(T)*. A central subtree with the minimum number of points is a least central subtree of a tree T. A least central subtree of *T* is the best possible connected substructures of *T* among all connected substructures.

We give some structural properties of a least central subtree of a tree. We describe exactly how the center and the centroid and a least central subtree of a tree are interconnected. Our main results prove that a least central subtree of a tree contains the center and at least one point of the centroid of the tree.

T1B.2

Polyhedron of Human Trees

Thanh Hai NGUYEN[*], Jean - François MAURRAS et Viet Hung NGUYEN

In this presentation, we study the convex hull of the rooted binary trees with *n* leaves. They are Human trees. Such a tree is mainly used to represent an alphabet of a written language. To a given Human tree with *n* leaves, we associate the point in the *n*-dimensional space, whose coordinates corresponding to the

leaf *i* is the number of edges between the leaf *i* and the root. For example, for an alphabet of 4 characters *{a, b, c, d}*, we have 13 possible Human trees whose corresponding points are:
(3, 3, 2, 1), (3, 3, 1, 2), (3, 2, 3, 1), (3, 2, 1, 3), (3, 1, 3, 2), (3, 1, 2, 3), (2, 3, 3, 1), (2, 3, 1, 3), (2, 1, 3, 3), (1, 3, 3, 2), (1, 3, 2, 3), (1, 2, 3, 3), (2, 2, 2, 2).
We will call SH_n the set of the points corresponding to all possible Human trees with *n* leaves. The optimization problem associated to this polyhedron is minimizing the linear function $\sum_{leaf\ i} w_i \times l_i$, where w_i is the weight of leaf *i*, and l_i is it's coordinated. There exists a greedy polynomial algorithm that solves this problem called the Human algorithm.
In this presentation, we are interested to the description of the convex hull of the points of SH_n, that will be noted PHT_n the Polyhedron of Human Trees in dimension *n*. We will recall also the Human algorithm and established links between facets in dimension *n* and those in dimension *n*+1. We describe several families of facets – defining inequalities of this polyhedron. Among them, there are a family of facets whose coefficients form a Fibonacci sequence. These inequalities also define facets for the dominant of PHT_n. In this case, we can apply the *0*-lifting technique to extend the Fibonacci family to a larger family of facet – defining inequalities. We show that this new family characterizes the points generated by the deepest trees.

References:

1. R. G. Gallager: *Information theory and reliable communications*, J. Wiley, 1968, pp. 45.
2. D. A. Human, *A method for the construction of minimum redundancy codes*, Proc. IRE, 40 (1951), pp. 1098 – 1101.
3. M. M. Kapranov: *The permutoassociahedron, Mac Lane's coherence theorem and asymtotic zones for the KZ equation*, (J. of Pure and Applied Algebra 85 (1993)). pp. 119 – 142.
4. Carl W. Lee: *The Associahedron and Triangulations of the n – gon*, (Europ. J. Combinatorics, 10 (1989)). pp. 551 § 560.
5. J. F. Maurras : *Programmation linéaire, complexité*, (Springer, 2002).
6. D. S. Parker et P. Ram: *The construction of Human codes is a submodular ('convex') optimization problem over a lattice of binary trees*, SIAM. J. Comput. 28(5), 1999, p. 2 – 5.
7. M. Pouzet, K. Reuter, I. Rival and N. Zaguia: *A generalized permutahedron*, Algebra Universalise, 34 (1995), pp. 496 – 509.

T1B.3

Coordinated supply chain scheduling

Tadeusz Sawik

The two approaches: monolithic and hierarchical, based on multi-objective mixed integer programming are proposed for coordinated supply chain scheduling. The supply chain consists of multiple manufacturers (suppliers) of parts, a single producer of finished products and a set of customers which generates the final demand for products. Each supplier has a set of identical production lines in parallel for manufacturing of parts, and the producer has a flexible assembly line for assembly of products.

The following static and deterministic supply chain scheduling problem is considered. Given a set of orders, the problem objective is to determine a schedule for manufacturing of parts at each supplier and for delivery the parts from each supplier to the producer, and to find a schedule for an assembly of products for each order by the producer, such that a high customer service level is achieved and the total cost of supply chain inventory holding, of the production line start-ups and of the part shipments is minimized.

In the monolithic approach, based on a weighted-sum program, the non-dominated schedules of manufacturing, supply and assembly are determined simultaneously. In the hierarchical approach based on a lexicographic optimization, first the maximal subset of orders that can be completed by customer requested due dates is found and for the remaining orders delayed due dates are committed to satisfy capacity constraints. Then, the non-delayed assignment of orders to planning periods over the horizon is found to minimize the holding cost of finished product inventory of the customer orders completed before their due dates. Finally, the manufacturing and delivery schedules of the required parts are determined to minimize the total cost of holding the inventory of parts, of start ups the production lines and of shipments the parts from the suppliers to the producer.

Numerical examples modeled after a real-world supply chain scheduling in the electronics industry are presented and some results of computational experiments are reported.

Keywords: Supply chain operations, Scheduling, Multi-objective integer programming.

T1B.4

Design and Analysis of a Modified Work Function Algorithm

Alfonzo Baumgartner, Tomislav Rudec, Robert Manger[*]

In this paper we study a modified work function algorithm (WFA) for solving the on-line *k*-server problem. Our modification is based on a moving window, i.e. on an approximate work function that takes into account only a fixed number of most recent on-line requests. The paper first gives a precise specification of the modified WFA. Then it explains how the considered algorithm can be implemented efficiently by network flows. Next, some experiments are presented, where the performance of the modified WFA is measured in terms of the incurred total cost of serving. The same experiments also estimate the computational complexity of the implemented algorithm, and give comparisons vs. the original WFA. The final part of the paper investigates the competitiveness of the modified WFA.

Keywords: on-line problems, on-line algorithms, k-server problem, work function algorithm (WFA), moving windows, implementation, network flows, experiments, performance, computational complexity, competitiveness.

T1C.1

New Bounds and Constraint Propagation Techniques for the Clique Partitioning Problem

Florian Jaehn[*] *and Erwin Pesch*

This paper considers the problem of clustering the vertices of a complete, edge weighted graph. The objective is to maximize the edge weights within the clusters (also called cliques). This so called Clique Partitioning Problem (CPP) is NP-complete, but it has several real life applications. E.g. a task that frequently arises in qualitative data analysis is to uncover natural groupings, or types, of objects, each of which is characterized by several attributes. One can think of these objects as vertices of an edge-weighted graph. Each positive or negative weight represents some measure of similarity or dissimilarity, respectively, of an edge-defining object pair. A clustering of the objects into groups is a partition of the graph. The set of edges connecting vertices of different subsets from some partition of the graph is called a cut. In order to find groups as homogeneous as possible, positive edges should appear within groups and negative edges in the cut. Hence, a best clustering is one with a minimal cut weight.

Numerous heuristic and exact approaches as well as benchmark tests have been presented in the literature. Most exact methods use branch and bound with branching over edges. In this presentation we propose new improvements for such an branch and bound algorithm solving this problem. Firstly, we present upper bounds that are based on the triangular restrictions, i.e. the fact that if vertices a and b are in the same cluster, and a and c are in the same cluster, then b and c have to be in the same cluster. The computation of these upper bounds can be done in polynomial time concerning the number of vertices. In all instances we tested, the initial upper bound at the root node was reduced by at least 60% (compared to the most common upper bound known from literature). Secondly, we use constraint propagation techniques. Those are based on the idea that at any node we can compute in polynomial time how much the upper bounds will change if a certain edge is fixed (i.e. the fact that two vertices must or must not be in the same cluster). Especially when the difference between upper and lower bound becomes small, these techniques lead to numerous fixations of edges. In two out of eleven instances tested, the upper bounds and the constraint propagation techniques (although easy to apply) lead to the optimal solution in the root node. Thirdly, we present a new branching scheme. From each node two new nodes derive. However, in general they do not halve the search space. Thus, our branching scheme guarantees that the child node containing the bigger part of the search space has a tight upper bound so that this part will be fathomed more quickly.

The theoretical improvements are reflected by computational tests with real life data. The runtime is decreased significantly compared to other publications.

T1C.2

On unicyclic graphs with a lower bounded girth

Walid Ben-Ameur, Makhlouf Hadji, Adam Ouorou*

Given a weighted undirected graph G=(V,E), we aim to compute a minimum weight subgraph G' =(V,E') with unicyclic connected components. Moreover, the number of edges of
each cycle of G' should not be less than p+1. Said another way, the girth of G' will be at least p+1.
This problem has some applications in telecommunications and vehicle routing. It has many connections to old problems. For example, when p = |V| -1, we get the traveling salesman problem. This clearly implies that our problem is also NP-hard.
If no girth constraint is imposed (p=2), the problem becomes easy. In fact, it is already known that the set of edge subsets inducing a graph where each connected component contains at most one cycle is a matroid. This matroid is generally called the bicircular matroid. Then a simple greedy algorithm can be applied to compute a minimum cost network where each connected component contains exactly one cycle.
It is also easy to compute a minimum cost unicyclic graph. This comes from the fact that the set of edge subsets inducing a graph containing at most one cycle is a matroid. This
matroid is obviously included in the bicircular matroid.
If the unicyclicity requirement is replaced by a cyclicity constraint (each connected component is exactly a cycle), we get the two-factors problem which is a well known easy problem. If a girth constraint is added (no cycles of length less than 5), the problem becomes difficult.
If the unicyclicity constraint is relaxed, we get another known problem: find a minimum (or maximum) weight graph not containing cycles of length less than p+1. This problem is equivalent to the covering problem where we look for a minimum (or maximum) weight set of edges intersecting all cycles of length than p+1. This problem is also NP-hard.
We will describe some valid inequalities for the design of unicyclic graphs with girth constraints. The faces induced by these valid inequalities are also studied. A cutting
plane algorithm bases on these inequalities is implemented to solve the problem. Some of these valid inequalities can be separated in polynomial time.

Keywords : Combinatorial Optimization, Network Optimization, Polyhedral Study, Cutting Plane algorithm, Unicyclic Graphs, Matroïds.

T1C.3

On cyclic properties of locally connected graphs with bounded vertex degree

Valery Gordon , Yury Orlovich , Chris Potts, Vitaly Strusevich*

We consider the existence of Hamilton cycles for locally connected graphs with a bounded vertex degree. It is well-known that the problem of deciding whether a given graph is hamiltonian, is NP-complete, and it is natural to look for conditions for the existence of a Hamilton cycle for special classes of graphs. Study of hamiltonicity conditions for graphs with a prescribed local structure is one of rather important directions in graph theory.

Graph G is *locally connected* if for any vertex u of G the neighborhood of u (i.e., the set of vertices adjacent to u) induces a connected subgraph of G. Let $\Delta(G)$ and $\delta(G)$ denote, respectively, the *maximum* and *minimum vertex degrees* of graph G. Graph G is *hamiltonian* if G has a *Hamilton cycle*, i.e., a cycle containing all vertices of G. A graph of order n is called *fully cycle extendable* if every its vertex lies on a triangle and for every cycle C of less than n vertices there exists another cycle C^* containing all vertices of C plus a single new vertex. Clearly, any fully cycle extendable graph is hamiltonian.

We explicitly describe all connected, locally connected graphs with $\Delta(G) \leq 4$. We show that every connected, locally connected graph with $\Delta(G) = 5$ and $\delta(G) \geq 3$ is fully cycle extendable which extends the results of P.B. Kikust (Latvian Math. Yearbook, 16, 1975) and G.R.T. Hendry (J. Graph Theory 13, 1989) on fully cycle extendability of connected, locally connected graphs with maximum vertex degree bounded by 5.

Furthermore, we prove that problem HAMILTON CYCLE for locally connected graphs with $\Delta(G) \leq 7$ is NP-complete (this result holds even for 7-regular locally connected graphs). As a consequence, the maximum integer Δ^* for which the HAMILTON CYCLE problem for locally connected graphs with $\Delta(G) \leq \Delta^*$ is polynomially solvable lies in the interval $4 \leq \Delta^* \leq 6$.

T1C.4

A surprising link between two invariants of graphs

Alain Hertz[*], *Pierre Hansen, Rim Kilani, David Schindl, Odile Marcotte*

With the help of the Graffiti system, Fajtlowicz conjectured around 1992 that the average distance between two vertices of a connected graph G is at most half the maximum order of an induced bipartite subgraph of G. We have proved a strengthening of this conjecture by showing that the average distance between two vertices of a connected graph G is at most half the maximum order of an induced cycle-free subgraph. This means that there is a strong and surprising link between the average distance in a graph and the minimum number of vertices that must be deleted from a graph to break all cycles.

T2A.1

Semi-Lagrangian Relaxation applied to the Quadratic Assignment Problem

Cesar Beltran-Royo[*], *Miguel Constantino, Huizhen Zhang*

In this work we apply the Semi-Lagrangian Relaxation (SLR) method (*) to solve the Quadratic Assignment (QA) problem. We propose a new Linear Programming (LP) formulation of the QA problem that is better suited for the SLR method compared to previous LP formulations. In the computational experiments on QA instances from the QAPLib web, the new formulation outperforms previous ones.

(*) The SLR method was successfully introduced in (Beltran06) to solve the p-median problem. Recently, the SLR method has solved some previously unsolved Uncapacitated Facility Location instances from the UflLib web. The SLR method is a modified Lagrangian relaxation. It is well known that, Lagrangian relaxation is commonly used in combinatorial optimization to generate *bounds* for the optimal cost. In contrast, the SLR method generates *optimal* integer solutions.

Keywords: Lagrangian Relaxation, Combinatorial Optimization, Quadratic Assignment (QA) problem.

T2A.2

Comparison of some mathematicals models for a Staff Rostering Problem

E. Naudin[*], P. Chan, M. Hiroux, T. Zemmouri, and G. Weil

We propose to compare the computational behavior of several mathematical models designed to solve the staff rostering problem. It consists of assigning workers to tasks whose starting and ending times are fixed. The novelty of this problem is that the constraints are defined over different horizons.

The first model uses assignment variables of person on tasks, these decision variables contain the least information. The other models use assignment variables on rosters with different horizons: firstly rosters on days and secondly, rosters on the complete horizon. They are obtained through a Dantzig-Wolfe decomposition of the first [2]. These models require column generation to be solved (see [1], [3] and [4]).

A solution to this problem must respect the following constraints:

The staffing requirement of each task is to be covered as completely as possible (1)

The set of assignments of each worker must verify:

– At all instants, each worker is assigned to at most one task; (2)

– Two consecutive working days are separated by a rest period with a minimal duration. (3)

– The total working time per day is limited to a maximal duration; (4)

– The total working time over the (weekly) time horizon is limited to a maximal duration; (5)

As we can see, the problem has constraints expressed over different horizons: short-horizon (1), medium-horizon constraints (2) and (4) and long-horizon constraints (3) and (5). The following table resumes the processing of each constraint in the three models:

Constraint	Model 1	Model 2	Model 3
Staffing requirements (1)	Master Problem	Master Problem	Master Problem
Resource Constraint (2)	Master Problem	SP: Graph	SP: Graph
Daily rest duration (3)	Master Problem	Master Problem	SP: Graph
Daily work duration (4)	Master Problem	SP: Capacity	SP: Capacity
Total work duration (5)	Master Problem	Master Problem	SP: Capacity

In our presentation, we will compare the computational behaviour of these models in different scheduling scenarios.

Key words. Staff rostering problem, Linear programming, Column generation

References

1. C. Barnhart, E. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance. Branch-and-Price: Column Generation for Solving Huge Integer Programs. Operations Research, 46:316-329, 1998.
2. G.B. Dantzig and P. Wolfe. Decomposition Principle for Linear Programs. Operations Research, 8:101-111, 1960.
3. M. Gamache, F. Soumis, G. Marquis, and J. Desrosiers. A Column Generation Approach for Large Scale Aircrew Rostering Problems. Operations Research, 47: 247-263, 1999.
4. B. Jaumard, F. Semet, and T. Vovor. A generalized linear programming model for nurse scheduling. European Journal of Operational Research, 107 (1) p.1-18, 1998.

T2A.3

Bi-Objective Integer Programming Approach to Optimal Assignment of Supporting Services in a Health-Care Institution

Bartosz Sawik

The supporting services have a strong impact on performance of health-care institutions such as hospitals. For example in a hospital, the supporting services include financial management, logistics, inventory management, analytic laboratories, etc. This paper presents a weighted-sum program for bi-objective

resource allocation among supporting services in health-care institution. The primary optimality criterion of the problem is to minimize operational costs of the supporting services and the secondary criterion is to minimize number of people required for the services availability. The problem constraints represent specific conditions for resource allocation in the health-care institution. The overall problem is formulated as bi-objective assignment problem, where the decision variables represent the assignment of people to various services. This paper proves practical usefulness of integer programming approach to optimization of supporting services in the health-care institutions. The results of some computational experiments modeled after a real data from Polish hospital are reported.

Key words: bi-objective integer programming, assignment problem, service operations management

T2A.4

A Weighted-Sum Mixed Integer Program for Multi-Criteria Portfolio Optimization

Bartosz Sawik

This paper presents the portfolio optimization problem formulated as a multi-criteria mixed integer program. An extension of the Markowitz portfolio optimization model is considered, in which the variance has been replaced with the Value-at-Risk (VaR). The VaR is a quantile of the return distribution function. In the classical Markowitz approach, future returns are random variables controlled by such parameters as the portfolio efficiency, which is measured by the expectation, while risk is calculated by the standard deviation. As a result the classical problem is formulated as a quadratic program with continuous variables and some side constraints. In the approach proposed in this paper the portfolio optimization problem is formulated as a multi-criteria mixed integer program. The main objective is to allocate wealth on different securities to maximize the weighted difference of the portfolio expected return and the threshold of the probability that the return is less than a required level. The auxiliary objectives are minimization of risk probability of portfolio loss and minimization of the number of security types in portfolio. The three types of decision variables are introduced in the model: a continuous wealth allocation variable that represents the percentage of wealth allocated to each security, a binary selection variable that prevents the choice of portfolios whose VaR is below the minimized threshold and a binary variable that represents number of securities in portfolio. The results of some computational experiments with the mixed integer programming approach modeled after a real data from Polish stock exchange are reported.

Key words: multi-criteria portfolio optimization, weighting approach, mixed integer programming, Value-at-Risk

T2B.1

On a class of Periodic Scheduling Problems

Michelon P.[*]*, Quadri D., Negreiros Gomes M. J.*

Periodic scheduling is concerned with processing, on a set of identical machines (or identical unitary resources), periodic tasks or activities over an infinite horizon. Each activity is characterized by its duration d. In a basic version of the problem, all these durations are equal to 1 and a period T is also associated with each of the task so that if task i is scheduled on time t then it must also be strictly scheduled on times t+d+T, t+2d+2T and so on. The problem consists then in minimizing the number of machines to process periodically the tasks. This basic problem has been shown as being NP-hard.

In this paper, we are concerned with a relaxation of the strict periodicity requirement. Two positive integer numbers, Fmin and Fmax, are associated with the tasks and correspond, respectively, to a minimum and a maximum delay for the repetition of the activity. Hence, if the k-th execution of a given task has been scheduled on time t then, its (k+1)-th execution must take place between t+d+Fmin and t+d+Fmax. This study was motivated by a logistic problem that occurs in the combat of the dengue, one of the major tropical disease.

We will present some general properties of the problem, an integer programming formulation for which we give the optimal solution of the linear relaxation, a trivial lower bound which is compared to the linear relaxation bound and a greedy heuristic.

T2B.2

Scheduling of multiprocessor tasks on dedicated machines by hypergraph coloring

Krzysztof Giaro, Marek Kubale[*]*, Paweł Obszarski*

We address a generalization of the classical 1- and 2-processor unit execution time scheduling problem on dedicated machines. In our chromatic model of scheduling, machines have non-simultaneous availability times and tasks have arbitrary release times and due dates. Also, the versatility of our approach makes it possible to generalize all known classical criteria of optimality. Under these stipulations we show that the problem of optimal scheduling of sparse tree-like instances can be solved in polynomial time. However, if we admit dense instances then the problem becomes NP-hard, even if there are only two machines in the system.

The problem of scheduling can be modeled as the cost list edge-coloring of hypergraphs in which vertices correspond to machines, hyperedges correspond to multiprocessor tasks, colors model time slots, and lists of colors stand for availability constraints. Since both problems are equivalent to each other, the optimal list cost edge-coloring of hypergraphs determines the optimal cost of scheduling.

T2B.3

Cutting planes for an industrial unrelated parallel machines problem with release dates and tails

Cédric Pessan[*]*, Meng Li, Emmanuel Néron*

The considered problem is the production resetting optimization problem in an industrial case found in SKF factories. A production resetting is an important phase that allows switching production from one product type to another: all machines of a production line have to be setup with new tools adapted to the new product type. These setup operations are processed by skilled operators. Depending on their experience, the operators need different amount of times to perform setup tasks. This problem can be identified as an unrelated parallel machine problem where the resources are the operators and the tasks are the setup tasks on each machine of the production line. The objective is to restart production as early as possible.

A setup task M_i can only be performed when the corresponding machine is empty: we need to wait a time r_i until the machine is empty: this is the time required for the last piece of the current lot to go from M_1 to M_i. Similarly, the production is measured at the end of the line, so we need to wait a time q_i at the end of each setup task M_i so that the first piece that is processed by M_i after restarting, reaches the end of the line. Thus, we know that the non decreasing r_i order of the tasks is the same as the non increasing q_i order. Therefore, the problem we study is the $R|r_i, r_i \geq r_j => q_i < q_j|\max(C_i+q_i)$. On each machine, the optimal order of the tasks is the non decreasing r_i order because there are no preemptions in the solution found by Jackson Preemptive Schedule.

A destructive lower bound for this problem can be computed by solving the decisional problem $R|r_i,d_i|$- (with $d_i=C_{max}-q_i$). This problem is NP-Complete but the preemptive relaxation is polynomial and can be solved using linear programming [Lawler,Labetoulle 1978]. In this program, the time is discretized in intervals I_k bounded by the set of all r_i and d_i.

In our work, we define cutting planes that we add to the linear program in order to improve the lower bound. These cutting planes try to partially take into consideration the non preemption constraints and the order of the tasks on each operator. To do this, we compute mandatory parts W_{ijk} of the tasks: this is the minimum time spent on task M_i during interval I_k if the task is assigned to operator O_j in the optimal solution of the non relaxed problem.

Using these mandatory parts, we adapt the cutting planes defined in [Mokotov,Chretienne,2002] for the R||C_{max} to our problem: in our problem, the cutting plane try to limit the load on each interval I_k by taking into consideration that there is no preemption in the optimal solution. We also define other cutting planes that detect assignment incompatibility using the mandatory parts: the detection is based on the order of tasks on each operator in the optimal solution. The improved lower bound is tested inside a Branch-and-Bound [Pessan et al., 2008].

Keywords : Scheduling, unrelated parallel machines problem, lower bound, cutting planes, preemptive relaxation.

T2B.4

Scheduling, Routing and Assigning Wavelengths to Lightpaths in Optical Networks

Nina Skorin-Kapov, Pablo Pavón Mariño*

In this work, we investigate the problem of scheduling lightpaths (i.e. transparent all-optical connections) in optical networks based on Wavelength Division Multiplexing (WDM) technology. Namely, due to the periodic nature of traffic, it may be possible to pre-define a schedule for establishing and tearing down lightpaths, as opposed to establishing static ones, in order to more efficiently utilize network resources. Once scheduled, establishing a lightpath involves routing it over the physical topology of the optical network and assigning a wavelength to each hop, subject to certain constraints. This is commonly referred to as the Routing and Wavelength Assignment (RWA) problem, often solved using a combination of routing and graph coloring techniques. We aim to tackle the scheduling and RWA problems simultaneously. This means searching for the temporal evolution of RWA schemes which efficiently adapt to the known traffic variations. RWA efficiency can be measured in terms of the number of wavelengths used; the maximum load (congestion) on any link in terms of lightpaths and/or traffic flows routed over individual lightpaths; and/or the total throughput. Since this problem is NP-complete, we develop heuristic algorithms to solve it suboptimally. Their performance merits are assessed through implementation and testing in the MatPlanWDM tool. MatPlanWDM is a MATLAB-based software publicly available at the MATLAB central web site. It is composed of an application kernel, a set of libraries of related algorithms and a graphical user interface. Its general goal coincides with its purpose in this work: to aid the implementation and evaluation of optimization algorithms for lightpath-based optical networks.

T2C.1

Machine layout for closed-loop conveyors

Feristah Ozcelik, A.Attila Islier*

Material handling facilities are the major factors that contribute to productivity of manufacturing environments. Manufacturing systems consist of machines/workstations arranged in some order and linked together by material handling systems. Among the several types of material handling equipment, conveyor is a widely used method for transporting and storing material between stages of production. In this paper we consider a closed loop conveyor system having multiple loading and multiple unloading stations. There is an input (output) queue for each loading (unloading) station. We assume that all input and output queues are of sufficient capacity so that loads are never blocked. Various job types (with user-specified job routing data) are processed in the system. Loads enter and leave the system through one of the machines. When a load is processed at machine i, it leaves the system if machine i is the last machine on its route. Otherwise, it joins the input queue at machine i and waits to be loaded on the conveyor to visit the next machine on its route. The conveyor is unidirectional. The problem is to assign n machines to n candidate locations around the loop. A Genetic Algorithm is developed to solve this machine layout problem. The effectiveness of proposed algorithm is evaluated with a small problem taken from literature.

T2C.2

On solving the multi-period location-assignment problem under uncertainty

Maria Albareda-Sambola, Antonio Alonso-Ayuso, Laureano F. Escudero[*],
Elena Fernandez, Celeste Pizarrro

Given a planning horizon, a set of customers and a set of facilities (e.g., production plants), the Multi-Period Stochastic Facility Problem, MPSFLP) is concerned with locating the facilities and assigning the customers to the facilities along given time periods in a time horizon. The aim is to minimize the facility setup cost, the customer assignment and service costs, and the expected penalty for not attending the customers demand whenever it is available. The pure 0--1 deterministic version of the problem, where it is assumed that the service time periods of the customers are known as well as the minimum number of customers to be served at each time period, is NP-hard}. Moreover, very frequently those parameters are uncertain.Therefore, the MPSFLP is an interesting application case of Stochastic Integer Programming. In the companion paper we have made an extensive computational comparison of three formulations for the deterministic version of the problem, namely, impulse based formulation, impulse-step based formulation and step based formulation. As a result the formulation based on the impulse-step approach has produced better results, so in this work we present its stochastic version.

We present a pure 0--1 Deterministic Equivalent Model} (DEM) for the multi-stage stochastic 0--1 MPSFLP with complete recourse, whose parameter's uncertainty is represented by a set of scenarios. We present an approach where the facility location and customer assignment along the time horizon do consider all scenarios without being subordinated to any of them. By considering a compact mathematical representation of DEM, we specialize the so-called Branch-and-Fix Coordination algorithmic framework. It exploits the structure of the model and, specifically, the non-anticipativity

constraints for the variables. The algorithm uses the Twin Node Family (TNF}) concept and the Lagrangean substitution bounding. Our procedure is specifically designed for coordinating the selection of the branching TNF, such that the non-anticipativity constraints are satisfied.

Keywords: Location-assignment, multi-stage stochastic 0-1 programs, branch-and-ix coordination

T2C.3

An integer model for solving a modification of the extended rapid transit network design problem

L.F. Escudero[*]*, S. Muñoz*

The extended rapid transit network design problem has recently been stated in the literature. Given a set of potential station locations and a set of potential links between them, this problem basically consists in selecting which stations and links to construct without exceeding the available budget, and determining an upper bounded number of noncircular lines from them, to maximise the total expected number of users.

In this work we deal with a slight modification of the above problem to allow the definition of circular lines. The main reasons for considering this modification are that it has been proved that circumferential configurations can increase the effectiveness of a rapid transit network, and, on the other hand, many cities have incorporated circular lines into their rapid transit networks.

A two-stage approach is presented for solving this problem. In the first stage an integer model is solved for selecting the stations and links to be constructed without exceeding the available budget, so that the total expected number of users will be maximised (without loss of generality, it has also been assumed that whichever two locations are linked by one line at most); it drastically reduces the dimension of a modification of a 0-1 model given in the literature to adapt it to our problem. In the second stage the

line design problem is solved by assigning each selected link to exactly one line, in such a way that the number of lines that go through each selected station is as small as possible; the proposed procedure requires no upper bound on the number of lines.

The optimisation engine CPLEX v11.0 is utilised for performing some computational experiments, which show that our approach also produces a drastic reduction on the computational effort required for solving the modification of the 0-1 model given in the literature.

Keywords: Station and link location; Circular line; Line designing; Degree of a node

T2C.4

Optimizing Assembly U-line Balance: an Industrial Case Study

Laurent CHAN, Felix T. CHAN, Van-Dat CUNG[*] and Gerd Finke

The U-line balancing problem belongs to the important family of assembly line balancing problems. The Simple Assembly Line Balancing Problem (SALBP) is well-known, coming historically from the automobile industry, in which a set of tasks with precedence relationships are assigned to a straight line of workstations. In SALBP, several objectives have been studied such as: (SALBP-1) minimizing the number of workstations for a given production rate, (SALBP-2) minimizing the cycle time (maximizing the production) for a given number of workstations and (SALPB-E) maximizing the line efficiency (minimizing the idle time of the line), when both cycle time and number of workstations are to be determined. All these problems are NP-hard, Becker and Scholl gave a survey in 2006.

In the context of Lean Manufacturing and Just-In-Time (JIT) production, the utilization of U-shaped lines, help to reduce the moving time of the operators and increase their interaction. The most important advantage of this layout is the flexibility to adapt the number of workers and stations according to the demands, i.e. the requested takt time. Miltenburg and Wijngaard in 1994 modeled the U-line Assembly Line Balancing Problem (UALBP). They proposed a dynamic programming procedure to solve small size instances (up to 11 tasks) and a heuristic to solve larger instances. In 1998, Urban presented an integer programming formulation of the problem which solves instances up to 45 tasks. Scholl and Klein in 1999 proposed the branch and bound procedure ULINO to solve instances up to 297 tasks. Our contribution is not to solve instances of larger size, but how to apply these results to a real industrial application in which we have to take into account the walking and waiting times of the operators, the number of operators and the size of buffers between the workstations.

Our industrial case comes from a nail gun factory, client of Dextus company. Their assembly U-line has usually seven workstations with seven units of capacitated buffers between them. An operator has to move, once he becomes idle, i.e. the upstream buffer of the current workstation is empty or the related downstream buffer is full. This means that the size of the buffers influences significantly the moves of the operators. The operators are assumed to be able to operate on any workstation. There are three optimization steps in the line design process: (step 1) determine the number of workstations given the maximal takt time, (step 2) balance the assembly line given the number of workstations and (step 3) determine the capacity of the buffers.

Since the industrial partner want to have a flexible solution in response to the variable demand, in step 1 the number of workstations is oversized regarding the maximal takt time requested. In steps 2 and 3, though it is possible to optimize the balance of the line and the capacity of the buffers with a large integer programming formulation which is difficult to solve, it is still difficult to capture the correct walking and waiting times of the operators with the moving rule in a deterministic model. We propose an approach coupling in serial an optimization model based on integer programming for step 2 with a simulation tool for step 3.

The simulation tool has been developed by Dextus to validate solutions before any implementation of the lines. This tool can handle many parameters such as the number of operators, the operating mode of the operators, their walking and waiting times, the buffer sizes and can generate different behavior of the lines.

For the step 2, we present an integer programming formulation close to the one proposed by Urban, but with a new linear objective function to balance the cycle time of each workstation with respect to the average of all the cycle times of the workstations. Our idea is to balance as much as possible the U-line with this optimization part which is presently handled by hand, and to test afterwards the solutions with the simulation tool to check their efficiency.

Empirical results in term of production pace, with increasing number of operators, with and without buffer operating modes will be presented to show that our solutions are better than the one presently implemented in the nail gun factory and the ones suggested by a human expert of the Dextus company.

T3A.1

Bilevel programming model for creating new measure of monetary policy

Darko Pongrac[*], *Kristina Šorić, Višnja Vojvodić Rosenzweig*

The basic central bank task is to achieve and to maintain price stability. For the purpose of achieving the formulated monetary policy, the Croatian National Bank (CNB) may use various measures and instruments. Croatian external debt has significant part of GDP for years, and CNB activities are mostly oriented to destimulate the growth of commercial banks external debt. Beside this, the goal is to slow down the households consumptions that are not based on the productivity growth and the growth of income. The main measure which is used for now with this purpose is marginal reserve requirements. In this paper the new measure with the influence on the slow down of households' indebtedness is considered.

The relationship between central bank and commercial banks is modelled as bilevel mixed 0-1 programming problem. The objective of the leader (in this case CNB) is to minimize the value of household's loans determining different percentages of the marginal reserve requirements. The objective of the followers (commercial banks) is to maximize the profit. This problem belongs to the class of NP-hard problems, and heuristics have been developed with the purpose of solving them. At the end, some ideas how to construct a heuristic based on tabu search are presented. We focus on the criteria of searching the neighbourhood that could be taken into the consideration, for example, the trade-off between decreasing the foreign loans' increase and increasing the interest rates. Some simulations are made to verify the model, and some computational results of these simulations are presented.

Key words: central bank’s measures, commercial banks credit activity, reserve requirements, bilevel mixed 0-1 programming problem, tabu search, heuristic

T3A.2

A tabu search heuristic for ship routing and scheduling with flexible cargo sizes

Jarl Eirik Korsvik, Kjetil Fagerholt[*], *Geir Brønmo*

This paper presents a planning problem faced by many shipping companies dealing with the transport of bulk products. These shipping companies typically have a certain amount of contract cargoes that they are committed to carry, while trying to maximize their profit from optional spot cargoes. Each cargo must be picked up in a given loading port and delivered in a given discharge port, and there are time windows in which loading and unloading must start. Hence, the problem is a version of the multi-vehicle Pickup and Delivery Problem with Time Window (m-PDPTW). However, in contrast to the m-PDPTW, the cargo sizes are flexible within an interval. Therefore, interwoven with the routing and scheduling decisions, the planner also has to decide the optimal cargo size for each cargo, as well as which spot cargoes to carry. A tabu seach heuristic, which embeds a specialized heuristic for deciding the optimal cargo quantities in each route, is proposed to solve the problem. Computational results show that the heuristic gives optimal

or near-optimal solutions to real-life instances of the problem within reasonable time. It is also shown that utilizing the flexibility in cargo sizes gives significantly improved solutions.
Keywords: Maritime transportation; Routing; Scheduling; Tabu search

T3A.3

Iterated Local Search algorithm for the constrained two-dimensional non-guillotine cutting problem

Selma KHEBBACHE, Christian PRINS, Alice YALAOUI*

The talk addresses the constrained two-dimensional non-guillotine cutting problem, which consists in cutting rectangular pieces from a large stock rectangle. The number of copies of each piece is limited by lower and upper bounds. Our objective is to maximise the total value of pieces cut. This problem can be classified as 2D-SLOPP (two dimensional single large object placement problems) in the typology of Wäscher et al. (2007), and has practical and important industrial applications (wood, glass…). It is well known to be NP-hard.

To solve this problem we propose an Iterated Local Search algorithm (ILS). To our knowledge, this method has never been used for this type of problem before. The ILS starts from a heuristic solution improved by local search. At each iteration, a copy of the current solution is perturbed and improved by the local search. If the new solution is better than the current solution, the next iteration starts with the new solution, otherwise the current solution is perturbed again.

To apply the ILS algorithm, we defined three procedures. The constructive heuristic of Alvarez-Valdes et al. (2005) is used to generate the starting solution. Perturbation which consists in a permutation between cut and uncut pieces allows generating new starting points for the local search. The local search procedure consists in removing cut pieces in random order and moves the remaining ones to the corners of the large rectangle.

We tested our algorithm on three sets of test problems and compared it with the GRASP algorithm of Alvarez-Valdes et al. (2005): the first group contains 21 problems from Beasley (1985), Hadjiconstantinou and Christofides (1995), Wang (1983), Christofides and Whitlock (1977), and Fekete and Schepers (1997). The length and weight of the large rectangle vary in [10,100], the number of pieces in [5, 30] and the maximum number of pieces which could be cut in [10, 97]. For all of them, the optimal solutions are known. The second set used by Leung et al. (2003) contains 10 problems. The length and weight of the large rectangle vary in [45, 400], the number of pieces in [5, 50] and the maximum number of pieces which could be cut in [10, 50]. The last set is composed of 630 larger problems generated by Beasley (2004). Our preliminary computational results are very competitive. They show that for the first set the average deviation to the optimum is 0.15% compared to 0.19% of the GRASP algorithm. Moreover, ILS improves two instances and does as good as the GRASP for 18 others. For the second set, ILS outperforms the GRASP for 4 instances and find the same results for 4 others. The mean deviation from the optimum is equal to 1.53% versus 2.05% for the GRASP. We are still testing on the remaining sets of problems and the current results are promising likewise.

Keywords: Cutting and Packing, Two-Dimensional Non-Guillotine Cutting, Heuristics, Iterated Local Search.

T3A.4

Heuristic for Matrix bandwidth minimization

N. Mladenović, D. Urošević, D. Perez-Brito, C. Garcia-Gonzalez*

The problem of reducing the bandwidth of a matrix consists of finding a permutation of rows and columns of a given matrix that keeps non-zero elements in a band as close as possible to the main diagonal. This NP-complete problem can also be formulated as a vertex labeling problem on a graph, where each edge corresponds to non-zero element of the matrix. For reducing the bandwidth of a matrix, we propose Variable neighborhood search based heuristic that successfully combines several recent ideas from the literature. Empirical results presented on the collection of 113 benchmark instances indicate that the proposed heuristic compares favorably to all previous methods. Moreover, with our approach, we found 26 new best solutions.

Keywords: Combinatorial optimization, Matrix bandwidth minimization, Metaheuristics, Variable neighborhood search.

T3B.1

Solving rich VRPs by column generation

Johan Oppen, Arne Løkketangen,Jacques Desrosiers*

Real-world routing problems are often represented by large and complex models, and instances of realistic size are very hard to solve. In most cases one cannot find optimal solutions, but has to use heuristic solution methods to find good solutions. It is, however, also of interest to find optimal solutions for smaller instances for comparisons with heuristic methods. We show that small instances of a real-world routing and inventory problem can be solved to optimality by using a column generation approach.

Our problem deals with transportation of animals for slaughter in Norway, and standard models for routing problems have to be extended to mirror important details from the real world. In this case, especially two aspects are interesting:

1. A solution (route plan) has to fit to both the slaughter plan and inventory capacities at the slaughterhouse. This means we add global inventory constraints to our routing problem.
2. When different animal types are transported on the same vehicle, they have to be kept in separate compartments on board the vehicle for animal welfare reasons. This leads to a loading problem, where the visiting order of the farms plays an important role in determining the total capacity of each vehicle.

Computational results and comparisons to heuristic methods are presented.

T3B.2

Integer programming formulations for vehicle routing problem with backhauls

*Barış KEÇECİ *, İmdat KARA*

Vehicle Routing Problem (VRP), which is one of the most important combinatorial optimization problems, is a key to efficient transportation management. Vehicle Routing Problem with Backhauls (VRPB) is a generalization of the ordinary Capacitated Vehicle Routing Problem (CVRP). In VRPB, the customers are divided into two subsets as “Linehaul Customers” where a given quantity of product must be delivered to each customer and “Backhual Customers” where a given quantity of product must be picked up from each customer. Distribution plan is built up such that the vehicles must visit the backhaul customers, if any, after all linehaul customers were served. As far as authors aware the first research on VRPB published by Deif and Bodin in 1984 and numerous number of studies has been published since then. The motivation of this study is the lack of two index polynomial size mathematical models of this type of problems. Two polynomial size mathematical models, one is *node based* and the other is *flow based*, are proposed in this paper. The computational experiments were done by the test instances taken from literature and randomly generated problems, using CPLEX 10.0. We compared the node based and the flow based formulation among them. At each instance we considered the CPU times for optimal solution for mixed integer program and the linear programming relaxation (LPR). In conclusion we observed higher LPR with node based formulation and smaller CPU time with flow based formulation. We suggest node based formulation to researchers who tend to develop mathematical programming based heuristic algorithms and flow based formulation to researchers who want to construct exact solution procedures for optimal solutions.

Key Words: Vehicle Routing Problem, Backhauling, Integer Programming

T3B.3

Solving a family of pickup and delivery vehicle routing problems arising in reverse logistics and the supply of oil and gas platforms

Irina Gribkovskaia[*], *Gilbert Laporte*

In one-to-many-to-one pickup and delivery single vehicle routing problems, a capacitated vehicle performs deliveries from a depot to a set of customers and also performs pickups at the same customers before returning to the depot. Several variants of the problem are considered, namely the general case, the case where customers have a limited capacity, the case where pickups are selective, the case where constraints are imposed on the lag time or sequencing order between the pickup and delivery visits at the same customer location, and the case where there are multiple commodities and time windows. Some of these variants have interesting applications in the supply of offshore oil and gas platforms and in reverse logistic. Formulating these problems as integer linear programs and solving them by a generic solver, such as CPLEX, enables the solution of only relatively small instances. It is however possible to solve all these variants by means of a unified tabu search heuristic. In some cases, one can prove that the percentage optimality gap can be quite small. In addition, one can show that it can sometimes be advantageous to perform two separate visits at the same customer location, one for the pickup and one for the delivery, rather than performing these two operations simultaneously.

Key words: vehicle routing; pickup and delivery; reverse logistics; tabu search.

T3B.4

Vehicle Routing for a Reverse Logistics Problem

Thierry Mautor[*], *Catherine Roucairol*

In addition to the classical distribution process of new products to the customers, the Reverse Logistics induces old products to be collected and transported in the reverse direction in order to be remanufactured, recycled or eliminated.

In this context, we have Vehicle Routing Problems with delivery and pick-up where vehicles both deliver the new (re)manufactured products and collect old products. These old products may have several possible destinations: remanufacturing unit, recycling unit, site for the elimination of waste products, ... We can remark that best solutions are then frequently non-hamiltonian, some customers having to be visited twice: one time for the distribution and one time for the collection.

After a presentation of the mathematical model, relations between the considered problem and other Vehicle Routing Problems are detailed. Finally, heuristic approaches are tested and compared on a large set of instances.

T3C.1

Efficient Method for Periodic Task Scheduling with Storage Requirement Minimisation

Karine Deschinkel[*], *Sid-Ahmed-Ali Touati*

This article addresses the problem of periodic storage optimisation in cyclic data dependence graphs (DDG), which is for instance applied in the practical problem of periodic register allocation for innermost loops on modern Instruction Level Parallelism (ILP) processors. The massive introduction of ILP processors since the last two decades makes us re-think new ways of optimising register/storage requirement in assembly codes before starting the instruction scheduling process under resource constraints. In such processors, instructions are executed in parallel thanks to the existence of multiple small computation units (adders, multipliers, load-store units, etc.). The exploitation of this new fine grain parallelism (at the assembly code level) asks to completely revisit the old classical problem of register allocation initially designed for sequential processors. Nowadays, register allocation has not only to minimise the storage requirement, but has also to take care of parallelism and total schedule time. In this paper, we do not assume any resource constraints (except storage requirement); Our aim is to analyse the trade-off between memory (register pressure) and parallelism in a periodic task scheduling problem. Note that this problem is abstract enough to be considered in other scheduling disciplines that worry about conjoint storage and time optimisation in repetitive tasks scheduling (manufacturing, transport, networking, etc.).

Existing techniques in this field usually apply a periodic instruction scheduling under resource constraints that is sensitive to register/storage requirement. Therefore a great amount of work tries to schedule the instructions of a loop (under resource and time constraints) such that the resulting code does not use more than R values simultaneously alive. Usually they look for a schedule that minimises the storage requirement under a fixed scheduling period. In this paper, we satisfy register/storage constraints early before instruction scheduling under resource constraints: we directly handle and modify the DDG in order to fix the storage requirement of any further subsequent periodic scheduling pass while taking care of not altering parallelism exploitation if possible.

In a previous paper we proposed an exact integer linear model for solving the problem of periodic scheduling with storage minimisation based on a theoretical approach (reuse graphs). Storage allocation is expressed in terms of reuse edges and reuse distances to model the fact that two tasks use the same storage location. Since computing an optimal periodic storage allocation is intractable in large data dependence graphs (larger than 12 nodes for instance) with a classical Branch and Bound method, we present here an efficient heuristic called SIRALINA . Our heuristic proceeds in two steps. A first optimal step provides scheduling variables and allows computing the potential reuse distances if the corresponding reuse edge is added. Then a second step solves a linear assignment problem using the Hungarian method in order to select the appropriate reuse edges.

Our practical experiments on many DDGs show that SIRALINA provides nearly optimal results for all periods in a satisfactory (fast) processing time. Its efficiency, on small problem instances, could be validated with respect to optimal solutions obtained from the exact integer linear model. We also present numerical results on large instances that could not be solved to optimality by an exact method. Consequently, SIRALINA is under inclusion inside a compiler for embedded systems.

Keywords: *periodic task scheduling, storage minimisation, heuristics, linear assignment*

T3C.2

An Arc Generation Approach for Vehicle Scheduling Problems

Stefan Bunte[*]*, Natalia Kliewer*

We consider the vehicle scheduling problem (VSP) arising in public transportation companies and its practical extensions like the well known multiple depot multiple vehicle type vehicle scheduling problem (MD-MVT-VSP) where the general task is to schedule vehicles of different vehicle types in order to cover a set of timetabled trips and to assign them to different depots.

Since a lot of research was done on this topic in the last years, we would like to give a short overview on proposed modeling and solution approaches for the mentioned problems. Besides we will discuss different real-world and artificially generated problem instances and how their different characteristics affect state-of-the-art solution methods.

We will present a new solution approach which is based on a time-space network model. The approach solves linear relaxations of multicommodity formulations on subnetworks and generates new arcs based on the reduced costs of the resulting optimal LP basis. By solving several iterations of the updated problem critical arcs of the network can be identified and fixed. Existing acceleration techniques like lagrangean pricing and LP stabilization were adjusted for the approach.

The approach not only leads to a faster solution of different VSP for real-life instances, but also can support planners in public transport companies by generating feasible solutions during the optimization process and therefore overcoming the so-called black-box behavior of optimization methods. Computational results will be presented on the basis of extensive experiments.

As a conclusion we want to point out what further research should be made in the future and discuss some basic concepts for applications in the field of operational and strategic planning in public transport.

T3C.3

Assigning Advertised Completion Times for Jobs with Uncertain Arrival Times

Singh G, Ernst A, and Robinson G*

Scheduling problems have traditionally been formulated as deterministic in nature. This is highly unrealistic for many industrial problems. For instance, the scheduling of maintenance tasks for a mineralogical plant ought to consider the large amounts of uncertainty about when staff with appropriate skills will be available and how long these tasks will take.

Online scheduling partially bridges the gap between deterministic and stochastic scheduling by providing scheduling policies. However, the scheduling at each decision point actually solves a deterministic problem. Simulation and scenario-based stochastic optimisation are two methods for solving scheduling problems which take the amount of uncertainty into account. However, they are unreliable unless large numbers of simulations are used.

In general, most computational effort is expended in evaluating the expected value of the objective function for a given schedule. In order to evaluate the objective function for a given schedule, we need to know the distributions of the completion times. All of these distributions, except the first one, require convolutions. This can lead to very difficult and intricate calculations.

We present a new approximation method, which can be used to evaluate a schedule for a stochastic scheduling problem. This method is approximate, but is attractive because it is computationally practical for large problems. This method is illustrated using a set of jobs that is to be scheduled on two parallel processors/machines, with precedence relations between jobs. The processing time for every job is the same and can be taken as the unit of time. Each job has a lognormally-distributed release date. The goal is to quote advertised completion times for all jobs. The criterion to be minimised is the total of advertised completion times plus weighted tardiness, where tardiness is defined as delay of actual completion times beyond advertised completion times.

The selection of the objective has been motivated by the fact that in many industries the likely completion times of jobs need to be communicated in advance. This is required for many reasons. In many circumstances, customers need to be told about likely completion times so that follow-up jobs can be scheduled accordingly. Parties who share a common resource need to inform each other as to when the resource will be next available.

We present our results on the comparison of the method presented in this talk with simulation and with discretizing the solution space. Preliminary results suggest that the approach presented in this talk is more accurate when the objective function is smooth. Further results from experimental runs will also be presented during the conference.

Keywords: uncertain release dates, parallel machines, precedence relations, stochastic scheduling.

T3C.4

Single Machine Scheduling and Due Date Assignment under Positional Deterioration

Valery S. Gordon, Vitaly A. Strusevich[*]

We study single machine scheduling and due date assignment (DDA) problems in which the processing time of a job depends on its position in a processing sequence. The jobs of set $N = \{1, 2,\ldots, n\}$ have to be processed without preemption on a single machine. The jobs are simultaneously available at time zero. The machine can handle only one job at a time. If the jobs are processed in accordance with a certain permutation $\pi=(\pi(1), \pi(2),\ldots, \pi(n))$, then the processing time of job $j = \pi(r)$, i.e., the job sequenced in the r-th position is given by $p_j^{[r]} = p_j g(r)$, where $g(r)$ is a function that specifies a positional deterioration effect and p_j is the normal or standard processing time. Here $g(1)=1$ and $g(r) \le g(r+1)$ for each $r,1 \le r \le n-1$. In particular, if $g(r) = r^A$ (where $A > 1$) we obtain polynomial positional deterioration, while if $g(r) = \gamma^{r-1}$ (where $\gamma > 1$) we obtain exponential positional deterioration.

In all problems that we consider in this paper, the jobs of set N have to be split into two subsets denoted by N_E and N_T. The jobs of subset N_T are essentially discarded, and a penalty α_j is paid for a discarded job $j\in N_T$. In a feasible schedule only the jobs of set N_E are sequenced, and each of these jobs is completed no later than its due date. The purpose is to select the due dates for the jobs and the sequence of the early jobs in such a way that a penalty function is minimized. One of these functions includes the cost of changing the due dates $\varphi(\mathbf{d})$ and the total penalty for discarding jobs $\alpha(N_T)$, i.e., $F_1(\mathbf{d}, \pi) = \varphi(\mathbf{d})+ \alpha(N_T)$. The other function $F_2(\mathbf{d}, \pi) = \varphi(\mathbf{d})+ \alpha(N_T) + E(N_E)$ additionally includes the total earliness $E(N_E)$ of the scheduled jobs, where the earliness of a job is the difference between its due date and its completion time. We look at two DDA models: (i) CON under which all jobs are given a common due date, and (ii) SLK under which the due date of a job is equal to its processing time plus a slack, common for all jobs.

We show that for any positional deterioration model, in an optimal schedule the scheduled jobs are sequenced in LPT order of their normal processing times. We develop dynamic programming algorithms that minimize the functions F_1 and F_2. The running time of our algorithms in $O(n^2)$ for the CON model and $O(n^3)$ for the SLK model.

Friday May 30, 2008

F1A.1

A parallel implementation of the Quadratic Three-Dimensional Assignment Problem using the Bob++ Framework

François Galea, Peter Hahn, Bertand Le Cun

We present a parallel implementation of a solver for the Quadratic Three-Dimensional Assignment Problem (Q3AP). This parallel implementation was built on top of the Bob++ library.

The Q3AP is a three-dimensional extension of the Quadratic Assignment Problem (QAP). Its subproblem lower bound is the three-dimensional Assignment Problem (3AP), which is a NP-hard problem. Our solver uses a polynomial lower bound of the 3AP.

Bob++ is an open-source framework for the implementation of parallel solvers based on tree-search algorithms such as branch-and-bound. Its supported parallel environments include POSIX threads for multi-core and multi-processor SMP machines, and Kaapi for large scale computation grids.

F1A.2

Reformulation of the Bilevel Knapsack Problem

Raïd Mansi, Luce Brotcorne and Saïd Hanafi[*]

Bilevel problems model hierarchical decision making process between two agents: the first one, called leader, fixes his decision taking into account the reaction of a second agent, called follower. Bilevel problems are modeled as optimization problems where some leader's constraints are the follower optimization program. Bilevel programs have been the subject of extensive study both from a theoretical and a practical point of view (see Colson *et al.* , for example). Bilevel programs are known to be generally intractable, mainly because they are generally non-convex and non-differentiable. Much attention has been devoted to the case in which both objective functions are linear. Even in this situation, the problem, which consists of obtaining a simple certificate of local optimality, is strongly NP Hard. In this paper we consider the bilevel knapsack problem (BKP), a linear bilevel problem with integer variables, where the follower's problem is a knapsack problem. BKP can be solved by applying generic algorithms such as those proposed by Bard and Moore and Moore and Bard. However, generic algorithms are unable to solve medium to large size instances, making it crucial to develop solution methods that rely explicitly on the knapsack structure of the follower problem (cf. Dempe and Richter, and Mansi, Brotcorne and Hanafi).

In this paper we propose an exact method with two phases to solve (GBKP) a more general BKP than the one considered by Dempe and Richter, and by Mansi, Brotcorne and Hanafi. GBKP is characterized by integer variables for the leader, binary variables for the follower, a knapsack constraint on the second level and general linear constraints on the first level. These last constraints are complicating ones since they can induce disconnected inducible region or infeasible one.

In the first phase of the algorithm we only consider the problem of the follower and we unaware the resources consumed by the leader in the knapsack problem. Then we use *the sparse dynamic programming* to obtain the set of all non-dominated feasible solutions of the follower problem. The rules of predominance used in this section are those proposed for the 0-1 knapsack problem and adapted by Dempe and Richter for the bilevel knapsack problem. We modified these rules taking into account the constraints of the leader, and then a solution should not be deleted except if it is strictly dominated. The application of the sparse dynamic programming results in a list *L* of all non-dominated solutions. The second phase of our algorithm consists in reformulating the *BKP* as *a linear integer program* where the feasible region is defined by the elements of *L.* They are sorted by ascending order compared to the consumed capacity of the knapsack and the profit of the follower. Numerical results are presented to show the efficiency of our method compared with those obtained by the algorithm of Moore and Bard.

Keywords: Bilevel Problem, Knapsack Problem, Dynamic Programming.

F1A.3

Mean and variance constraints, valid inequalities and Lagrangean relaxation for stochastic network design problems

Pablo A. Miranda

A general methodology is proposed to tackle non-linear combinatorial optimization problems, which deal with network design problems arisen in logistics and supply chain management. A common factor of the analyzed problems is the existence of nonlinear constraints and cost terms, explicitly involving demand mean and variance for each node of the network.

A proposed solution approach is based on lagrangian relaxation, decoupling network design variables (facility setting, customer assignments, and flows), from dependent variables of demand mean and variance. Additionally, valid inequalities for mean and variance variables are incorporated on formulations, resulting in stronger sub-problems and consequently reducing duality gaps. The methodology is analyzed into three different network design problems: i) facility location with inventory planning and capacity constraints, ii) stochastic capacitated clustering problems, and iii) transportation network design with inventory planning. Successful computational application for the first two problems are presented.

Keywords: Inventory-Location Models; Customer Clustering; Network Design Models; Lagrangian Relaxation; Valid Inequalities; Non-Linear Combinatorial Optimization.

F1A.4

Primal and convex hull relaxations for nonlinear integer programming problems

Monique Guignard[*]*, Aykut Ahlatcioglu*

Obtaining strong bounds for nonlinear integer programming problems has always been a challenge. Lagrangean relaxation is usually not appropriate, since the subproblems are still nonlinear integer problems, and unlikely to be easier to solve than the original problem. Relaxations based on solving linear integer problems are more likely to be successful.

The primal relaxation introduced by Guignard in 1994 relies on solving linear integer programming problems over convex hulls of integer solutions feasible for a subset of the constraints. Solution algorithms are based on penalty or augmented Lagrangean methods and either Frank and Wolfe or simplicial decomposition. They have been studied by Contesse and Guignard and implemented by Ahn, Ahlatcioglu and Guignard.

When all constraints are kept in the subproblems, one obtains the so-called convex hull relaxation or CHR (Albornoz, independently rediscovered by Ahlatcioglu). This relaxation will only work if one can solve relatively quickly linear counterparts of the original nonlinear integer problems. If these linear problems have the integrality property, one will only obtain the continuous relaxation of the original problem.

Since both primal and convex hull relaxations rely on Frank and Wolfe or on simplicial decomposition, convergence of the process to a valid bound requires convexity of the objective function. If the function is nonconvex but quadratic, and the variables are all 0-1, convexification is possible via SDP optimization (Billionnet, Elloumi and Plateau for instance). The bounds obtained via convexification have so far been disappointing; however they allow the otherwise impossible solution of moderately sized problems via cplex.

An advantage of CHR is that all constraints remain in the subproblems, thus all integer solutions found during the algorithm are feasible. This makes CHR an attractive primal heuristic. In particular it can be used for nonconvex as well as convex problems.

We will present results for several types of quadratic assignment problems, both convex and nonconvex.

F1B.1

A Mathematical Solution to the Cournot Model of Duopoly

Mehmet Can

Negotiating about the shares is costly, and the pie may decay or disappear if the negotiations go on for very long. At least since Edgeworth (1881) bargaining has been perceived as an important question in economics and political science. Nash (1950, 1953) used both the cooperative or axiomatic approach and the noncooperative one in his work on bargaining; he first characterized the unique outcome satisfying a set of axioms, and then proposed a non-cooperative game whose equilibrium was precisely this outcome. However, Nash's noncooperative model assumed that players had only one chance to reach an agreement, and that if they failed to do so they were unable to continue negotiating. This game seemed too simple to capture the richness of bargaining, and the noncooperative approach to bargaining received little attention until the 1970s.

Cournot (1838) anticipated Nash's definition of equilibrium by over a century, but only in the context of a particular model of duopoly. Not surprisingly, Cournot's work is one of the classics of game theory; it is also one of the cornerstones of the theory of industrial organization. We consider a very simple version of Cournot's model and variations on the model. We illustrate: (a) the translation of an informal statement of a problem into a normal-form representation of a game; (b) iterated elimination of strictly dominated strategies, and (c) a mathematical analysis approach to the solution of the problem.

F1B.2

An integration of game theory into the operations research

Hüseyin Hayri Nuroglu[*], *Elif Nuroglu*

Making decisions in today's business environment is a complicated and risky task. To make decisions easily and to increase the quality of decisions experts are trying to find different methods. Game theory is one of the most helpful tools that helps us to define competitors, possible strategies of each partner in the game and also expected payoffs for each player. The applications of game theory can be found in different fields from international relations, marketing, and business negotiations to oligopolies and political campaigns.

On the other hand, as the size and complexity of organizations increase it has become a more difficult job to allocate available resources into different activities of an organization. Managing the resources and activities of an organization by using a scientific and strategic way is called as operations research. By defining the goals or problems mathematically and using the available data, operations research shows us how to get a best or optimal solution.

In this article, the integration of game theory into the operations research is studied. Starting with classical ways in game theory of defining problems, strategies and payoffs, firstly solutions in terms of game theoretic approach are found. In the second step, game theoretic definitions are transformed into the language of operations research. In the last part of the article, this integration was explained with the help of hypothetical examples. It is worth to explain that problems in business environment can comfortably be solved by appropriate software packages such as LINGO and EXCEL-SOLVER. Moreover, for these applications not a high level of mathematics is required, therefore they are applicable for every decision maker from various fields. In contrast to general opinion about the complexity of formulating real life problems and solving them by using above mentioned software packages, this article shows that it doesn't require a very high level of expertise to apply everyday business problems in operations research.

F1B.3

On Fairness in Fair Queuing

Wieslaw Kubiak

The requirement of a fair bandwidth allocation to flows has been a recurrent topic in the design of fair queuing algorithms. These algorithms are widely considered to result in fair and stable networks. The recursive min-max fairness criterion as well as the absolute fairness and the relative fairness have been used in the design of the fair queuing algorithms. The absolute fairness is based on the maximum difference between the allocations provided by a given algorithm and those provided by the generalized processor sharing ideal, but infeasible, allocation algorithm. The relative fairness is based on the maximum difference between the allocation received by any two flows according to a given algorithm.
This presentation shows that the recursive min-max criterion is closely linked to the well known parametric apportionment methods. It also shows that the relative fairness is linked to another apportionment method which however is not house monotone. Finally, the presentation shows that the absolute fairness is in fact the minimization of the bottleneck of a just-in-time sequence.

F1B.4

Gap Minimization in Competitive Fair Division

Shao-Chin Sung, Milan Vlach[*]

We are concerned with a recently proposed model for competitive fair division. An informal description of the problem can be presented in terms of the so-called housemates problem as follows. Given

- a house with *n* rooms to be rented by *n* housemates,
- a positive total rent for the house,
- for each room, a nonnegative bid from each housemate, where the sum of all bids from each housemate is equal to the total rent for the house,

assign the rooms to the housemates and determine room rents in such a way that

- each housemate is assigned exactly one room,
- each room rent is nonnegative,
- the sum of all room rents is equal to the total rent for the house,
- the assignment of rooms together with room rents has a minimum gap where the gap is the difference between the greatest surplus and least surplus of housemates with respect to their bids.

We prove constructively that a solution whose gap is zero always exists and propose an efficient algorithm for finding such a zero-gap solution. However, among the zero-gap solutions, there may be such that each housemate receives a negative surplus. We show that if the negative surpluses are forbidden then the problem of the gap-minimization becomes NP-hard.

F1C.1

Inverse Model of the Flow Shop Scheduling Problem

Peter Brucker, Natalia V. Shakhlevich[*]

Inverse optimization has attracted much attention of researchers recently. Unlike traditional optimization models for which all parameters are given and the objective is to find the best solution that satisfies specific constraints, in inverse optimization the exact values of some parameters are unknown and they should be determined so that a prespecified solution becomes optimal.

In this paper, we study the inverse counterpart of the two machine flow shop scheduling problem. In the forward flow shop problem, each job of a given set should be processed first by machine A and then by machine B; the processing times are given and they are fixed. The objective is to find a job sequence that minimizes the makespan. The problem is solvable by the famous algorithm due to Johnson (1954).

In the inverse flow shop problem, a target job sequence, which is the same for both machines, is given together with the typical values of processing times. The objective is to modify the processing times within certain limits to be as close to the typical ones as possible so that the target job sequence becomes optimal.

The Johnson sequence is sufficient but not necessary for a job sequence to be optimal. We formulate the conditions that characterize the whole class of optimal job sequences for fixed processing times. Based on these conditions, we propose a mathematical programming formulation of the inverse flow shop problem for fixed processing times establishing NP-hard and polynomially solvable cases.

F1C.2

Two-machine flowshop scheduling problem with penalization of overlapping

Ameur SOUKHAL[*] and Christophe LENTE

An extension of the two-machine flowshop model that arises in industrial applications is considered. Contrary to the classical flowshop, the processing times are variable and increase in situation of overlapping.

In this study, *n* jobs, ready at time zero, are scheduled without preemption in order to minimize the makespan. Each job consists of two operations to be processed on the first machine then on the second machine in this order.

The job overlapping is permitted. According to the literature, the considered scheduled problem is defined as follow: the second operation is allowed to start before the first operation has been finished. Moreover, processing times of the operations on the second machine tend to increase with the amount of overlapping. For each job, the maximum amount of overlapping is given which is less or equal to the processing time on the first machine.

Thus, the second processing time is defined as a non-decreasing penalization function of the amount of overlapping. The complexity status of this scheduling problem is still open. We propose the following contribution: We prove that this scheduling problem is strongly NP-hard, by reduction from the known NP-hard 3-partition decision problem.

To solve the two-machine flowshop scheduling problem with the penalization of the overlapping, we propose a heuristic with a ratio 2, based on the Johnson's rule. We also show that the absolute deviation between the heuristic and the optimal solution is not greater than the large value of the maximum amount of authorised overlapping. In this case, the bound is tight.

Keywords: Scheduling; Flowshop; Overlapping; Complexity; Heuristic

F1C.3

Scheduling a Maintenance Activity on a Two-Machine Flowshop

Gur Mosheiov[*], Assaf Sarig

Scheduling a maintenance activity is an important decision in many production systems. We study a problem of scheduling a maintenance activity on a two-machine flow-shop. We consider three settings: (*i*) only the first machine needs to be maintained, (*ii*) only the second machine needs to be maintained, and (*iii*) both machines must be maintained simultaneously. As in many real-life situations, the maintenance must be performed within a given time interval. The objective is minimum makespan. All three problems are shown to be NP-hard. We introduce efficient heuristics which are shown numerically to perform consistently very well.

F1C.4

On cyclic identical coupled-task problems

Vassilissa Lehoux-Lebacque, Nadia Brauner, Gerd Finke[*]

Coupled-tasks were introduced by Shapiro in 1980 for scheduling the operations of a radar. The radar emits a pulse that is transmitted to a target and reflected back to the radar, which receives the pulse. Hence the radar must process two operations per task (emission and reception), and those operations are separated by a fixed duration. The coupled-task scheduling problem is then to process tasks of this type on a single machine (the radar). The objective is to minimize the makespan in the non-cyclic case or the throughput rate in the cyclic case.

Coupled-task problems have been studied widely for their various fields of application and for their theoretical interest. Orman et al. consider multifunction radars, used in a military context for weapon guidance, target searching and tracking. Gupta proposes other applications in the chemical industry or for the scheduling of workstations that require the refixturing of the parts between operations. The authors have recently added a new application of this model. They show that it is equivalent to a certain no-wait robotic cell problem. In this context, the scheduling of cyclic identical coupled-tasks is particularly relevant.

Orman and Potts have described the complexity status. The problem is difficult for general coupled-tasks. There are some polynomially solvable very special cases. However, the identical coupled-task case remains open. In this paper, we consider the cyclic identical coupled-task problem. Notice that the input of this problem consists of only three integers: the duration of the first operation, the duration of the second operation and the fixed distance that separates both operations. Hence, we have to deal with a high-multiplicity scheduling problem for which it may be difficult to prove that it belongs to NP, since a description of a schedule (giving for instance the starting times of the tasks) is not polynomial in the input size. Here, we describe a large class of feasible schedules, for which it is possible to find, in polynomial time, an optimal solution, a compact description of its schedule and its cycle time. There are strong indications that this class captures in fact all "interesting" schedules, which would imply, in the affirmative case, that the cyclic identical coupled-task problem is polynomially solvable.

F2A.1

Estimating the Cost of Deploying Teams to Sites in a Reconfiguration Process

Belén Melián, Manuel Laguna, Thomas Vossen*

We address the problem of estimating the cost of deploying teams of individuals capable of performing specialized task associated with the transformation of multiple sites belonging to a large organization. The motivation for studying this problem originates from the effort undertaken by the Transportation Security Administration (TSA) of federalizing 426 airports in the aftermath of the 9-11 terrorist attacks in the United States.

The problem of estimating the cost of deploying teams to reconfigure airports can be formulated as a mixed-integer program. In our formulations, we make the following assumptions: 1) The decision maker would like to minimize total cost, consisting of a cost per (fulltime-equivalent employee) FTE and a traveling cost that depends on both FTEs and distance traveled; 2) The conversion of all airports must be completed by given time horizon; and 3) There is a known FTE requirement and processing time at each facility.

The problem considered in this work is then to determine the number and size of teams and the deployment schedule in order to reconfigure all facilities within the timeframe and at a minimum cost. The cost function includes both traveling and labor costs. To the best of our knowledge, this problem has not been explicitly described and formulated in the operations research literature.

In order to solve this problem, we propose mathematical models and solution methods that are general and can be applied to similar situations where a reconfiguration is necessary in a fairly large number of non-homogenous facilities. We design a heuristic approach and then the computational experiments are performed with data for the airport-federalization problem.

F2A.2

Stowage planning in maritime bulk shipping

Lars Magnus Hvattum, Kjetil Fagerholt, Vinícius Armentano*

We introduce a problem arising in maritime bulk shipping. While the major decision is to plan vessel routes in order to service requests for the transportation of bulk cargos, we focus on the important subproblem of deciding whether a given vessel route is feasible with respect to stowage. Various constraints could possibly render a route infeasible, such as the physical capacity of the tanks onboard the vessel, specific regulations for transportation of hazardous materials, or requirements of the weight and stability of the vessel. We formulate a mixed integer programming model and discuss potentially useful objective functions. The problem is NP-hard, but must be solved repeatedly as a subproblem when designing ship routes. The potential of using standard MIP-solvers and specially tailored heuristics is discussed.

F2A.3

On a dual network exterior point simplex algorithm for the minimum cost network flow problem and its empirical behavior

Geranis Georgios, Paparrizos Konstantinos and Sifaleras Angelo[*]

Combinatorial Optimization is a large part of Mathematical Optimization. The Minimum Cost Network Flow Problem, (MCNFP) consists a wide category of network flow problems; perhaps one of the most important of the research area of Combinatorial Optimization. Recently a new Dual Network Exterior Point Simplex Algorithm (DNEPSA) for the MCNFP has been developed. This algorithm belongs to a special category of “exterior point simplex type” algorithms. Similarly to the classical network dual simplex algorithm (NDSA), the new algorithm starts with a dual feasible tree-solution and after a number of iterations, it produces a solution that is both primal and dual feasible, i.e. it is optimal. However, contrary to the NDSA, the new algorithm doesn't always maintain a dual feasible solution. Instead, after each iteration, it produces a new tree-solution which can be non-dual feasible and at the same time non-primal feasible. In order to do that, the algorithm keeps in touch with the dual feasible area and, iteration by iteration, it gets closer to the primal feasible area by reducing the solution's infeasibility. In this paper a preliminary computational study on the algorithm’s performance on randomly generated problems is analytically presented.

Keywords: Operational Research, Combinatorial Optimization, Minimum Cost Network Flow Problem, Exterior Point Simplex Algorithm.

F2A.4

Bounds and algorithms for the Bin Packing Problem with Conflicts

Albert E. Fernandes Muritiba, Manuel Iori, Enrico Malaguti, Paolo Toth[*]

In the “Bin Packing Problem with Conflicts” (BPPC), we are given a set V of items, each having a positive weight, and an infinite number of identical bins having a weight capacity C. We are also given a “conflict graph” G = (V, E), where E is a set of edges such that edge (i, j) belongs to E if an only if items i and j are in conflict. The aim of the BPPC is to assign all the items to the minimum number of bins, such that the total weight of the items assigned to a bin does not exceed the bin capacity C, and no bin contains items in conflict.

The BPPC is important because of the high number of real-world applications, and because it generalizes other important NP-hard combinatorial optimization problems. Real-world applications of the BPPC include examination scheduling, assignment of processes to processors, and load balancing of tasks in parallel computing. It also concerns particular delivery problems, such as food distribution, where some items cannot be placed in the same vehicle.

The BPPC generalizes both the “Bin Packing Problem” (BPP) and the “Vertex Coloring Problem” (VCP). The BPP is a special case of BPPC arising when no item is in conflict with the other items (i.e., when the edge set E is empty). The VCP is a special case of the BPPC arising when all the items have weight equal to zero (i.e., when the bin capacity C is infinite).

Several combinatorial lower bounds and bounding procedures based on the Linear Programming relaxation of different Integer Linear Programming formulations of the BPPC are proposed. Greedy

heuristic procedures and evolutionary and Tabu Search metaheuristic approaches are presented. An exact branch-and-price algorithm, based on the “Set Covering” formulation of the BPPC, is also proposed.

Extensive computational experiments on benchmark instances from the literature are reported.

Keywords: Bin Packing Problem, Vertex Coloring Problem, Combinatorial Optimization.

F2B.1

A New Neighborhood and Tabu Search for Job Shop Scheduling with Blocking Constraints

*Andreas Klinkert**, *Heinz Groeflin*

The Blocking Job Shop is an extension of the classical job shop problem that arises for instance in manufacturing environments where there are no intermediate buffers between machines. The absence of buffers implies blocking constraints in the sense that a job has to stay on a machine (and is blocking it) until the next machine becomes available and the job can be transferred for further processing.

The problem considered here is a generalization of the Blocking Job Shop (BJS) and shall be called the Generalized Blocking Job Shop (GBJS). It includes the following additional features: (i) it takes into account transfer times for moving a job from one machine to the next machine, and (ii) it allows for sequence-dependent setup times between consecutive operations on a machine. The need for taking into account transfer times comes from applications where transferring a job between two machines requires some specific parallel handling on both machines, involving a handover step on the first machine and a takeover step on the second machine, which must be synchronized and are assumed to have the same duration. Note that BJS is a special case of GBJS where all transfer times are zero and there are no setups between operations.

Solving job shop problems with blocking constraints poses some challenges since several structural properties found in the classical job shop (JS) are no longer present. To illustrate differences, two such properties shall be mentioned her, both being related to feasibility issues. First, in contrast to JS, a feasible partial schedule cannot always be extended to a feasible complete schedule (the authors showed that deciding whether this is possible is NP-complete). As a consequence, any heuristic that incrementally builds up a solution (e.g. based on priority rules) risks to run into infeasibility. Second, it is not straightforward to construct feasible neighbor solutions in a local search approach since moves based on simple swaps of adjacent operations typically yield infeasible schedules (while in the JS case, it is well known that swapping critical adjacent operations leads to feasible neighbors).

The main part of this article is dedicated to the construction of a new local search neighborhood for the GBJS problem which allows moving from a feasible solution to a new feasible solution. First, a disjunctive programming formulation for the GBJS problem is given, together with an associated disjunctive graph representation. The disjunctive graph concept for GBJS differs from the classical disjunctive job shop graph in several features. Most importantly, each operation is represented by two nodes (corresponding to the takeover and handover action), and the constraint that no two operations on a machine may overlap, leads to disjunctive sets consisting of a pairs of arcs with different extremities.

The mechanisms developed for constructing feasible neighbor solutions are essentially based on the following ideas: In order to reduce the makespan of a given feasible solution, at least one critical disjunctive arc has to be exchanged, corresponding to a swap of two adjacent operations on a machine. Since exchanging a single arc typically leads to an infeasible solution in GBJS, some additional disjunctive arcs have to be exchanged in order to maintain feasibility. A concept is therefore needed to identify subsets of disjunctive arcs (containing at least one critical arc) such that, when exchanging all arcs contained in such a subset, the resulting solution is again feasible.

Two structural properties concerning disjunctive GBJS graphs are elaborated which are essential for the neighborhood design. First, a closure concept is developed which relates to the problem of determining all disjunctive arcs that are "implied" by a given partial feasible selection of disjunctive arcs. Second, a key theorem on "short cycles" in disjunctive GBJS graphs is established which basically shows

that for any cycle of positive weight generated by an infeasible solution, there exists a "short cycle" of positive weight "visiting" each job at most once.

Based on these results, a neighborhood is developed and a corresponding tabu search algorithm for the minimum makespan GBJS problem is devised. Extensive computational experiments document the effectiveness of the approach and show that the solutions found outperform most of the benchmark results reported in the literature.

F2B.2

A Tabu Search Approach to an Urban Transport Problem in Northern Spain

Joaquín Pacheco[*], Silvia Casado, Ada Alvarez, José Luis González-Velarde

In this work we analyze an urban transport problem that the City Council of Burgos, a city in northern Spain, has posed to the authors. Given a fleet of buses and drivers, the problem consists in designing routes and assigning buses to the routes such that the service level is optimized. The optimality of the service level is measured in terms of the waiting time at the bus stops and the duration of the trip. Thus, the problem comprises of two decision levels (route design and bus assignment) and differs from other urban transport models found in the literature. In order to solve the problem, we propose two algorithms: one with a local search strategy and another with a tabu search strategy. In both cases, the solutions of the two decision levels are modified in alternating steps. The proposed algorithms obtained significantly better results than the tools currently applied by the transport authorities. In addition, the solutions obtained are very robust with respect to variations on demand, as shown by the experiments.

F2B.3

A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem

José Brandão

In this paper we study the *heterogeneous fixed* fleet routing problem (HFFVRP), which is a variation of the classical vehicle routing problem (VRP), containing several types of vehicles where each type has a different capacity and a different variable cost per distance unit travelled. In the HFFVRP the number of vehicles of each type is *fixed* and known in advance. The HFFVRP differs from the *heterogeneous* VRP because in this case the number of vehicles of each type is unlimited. The objective of the problem is to define a set of vehicle routes, each starting and ending at the depot, such that all customers' requirements are satisfied and in order to minimize the total cost. The HFFVRP is a NP-hard problem and, therefore, only approximate algorithms are suitable for solving practical problems.

The tabu search algorithm (TSA) is based on three types of neighbourhood move: insertion, double insertion and swap. In an insertion move, a customer is removed from its current route and a trial insertion is made in any other route, containing at least one of its δ-nearest neighbours, between any two other customers or the depot. In the double insertion, two customers are moved instead of one. The swap move is similar. Two candidate customers are selected from two different routes and interchanged and their place in the destination route is defined by the least insertion cost. These three types of move are not applied in every iteration, but instead they are applied alternately with a given frequency.

The trial move chosen depends on the effect of the move on the objective function. This objective function includes the variable costs (travelling costs) and a term that penalizes infeasible solutions in terms of capacity, which are also allowed.

At each iteration in the TSA, the best admissible solution from the candidate moves is selected. At the end of each iteration, the GENI procedure, developed by Gendreau, Hertz and Laporte, is applied to the two individual routes that have been changed to try to reduce their costs further.

In order to evaluate the performance of our algorithm we used a set test problems proposed by Taillard.

F2B.4

A hybrid Tabu Search / Branch & Bound approach to solving the Generalized Assignment Problem

Andrew J. Woodcock, John M. Wilson[*]

A new approach for solving the generalized assignment problem (GAP) is proposed that combines the exact Branch & Bound approach with the heuristic strategy of Tabu Search (TS) to produce a hybrid algorithm for solving GAP. The algorithm uses the mathematical programming software Xpress-MP to solve sub-problems generated by the TS guiding strategy. The TS approach makes use of the concept of referent domain optimisation and introduces novel add/drop strategies. In addition, the linear programming relaxation of GAP, that forms part of the Branch and Bound approach is itself helpful in suggesting which variables might take binary values. Computational results on benchmark test instances are presented and compared with results obtained by the standard Branch & Bound approach and also several other heuristic approaches from the literature. The results show the new algorithm performs competitively against the alternatives and is able to find some new best solutions for several benchmark instances.

F2C.1

Optimization Methods for Spam Filtering

Didier Colin[*], *Catherine Roucairol, Ider Tseveendorj*

Over the last few years, spam-filtering techniques have reached a satisfying maturity both in terms of accuracy and adaptation capabilities. Classification techniques, which are used to induce efficient filters, are now fully mastered, and it seems that present classification models offer little room for more improvements. In order to bypass this state of the art and bring more improvements to spam filters, we investigate how one could use optimization methods to attune filters to the specific issues and constraints found in the spam filtering area. Among those issues is the need to modelize and manage spammers strategies to delude the filters. In this presentation, we propose a selective learning scheme designed to maximize learning efficiency. In an offline context, this model uses a simple metaheuristic approach to select a subpart of training data such that the filter induced on that part maximizes its accuracy over the entire training set. In an online context, we show how one filter can discard incoming messages in order to prevent its knowledge base to be biased by messages which are not good representatives of their class and thus, may lead to a decrease in accuracy. We show that this approach synergizes well with existing classification models while increasing significantly their efficiency over time. More importantly, we show that this model make existing filters less vulnerable to spammers' attempts to delude their classification model.

F2C.2

Fuzzy C-means model and algorithm for data clustering

Sadina Gagula-Palalic[*], *Mehmet Can*

Pattern recognition has become a very important field over the last decade since automation and computerization in many systems has led to large amount of data being stored in the databases. The primary intention of pattern recognition is to automatically assist humans in analysing the vast amount of available data and extracting useful knowledge from it. Many algorithms has been developed for many applications, especially for a static pattern recognition. Since the information of theese processes can be non-deterministic over the time period, fuzzy approach can be applied to deal with this. In this work, fuzzy approach for optimization techniques in the pattern recognition will be developed. It will show a fuzzy model for data clustering and feature extraction that best suits for the process of pattern recognition when we deal with non-crisp data.

F2C.3

Heuristics for the design of maximum hydropathic complementarity molecules

Alberto Ceselli, Sandro Fornili, Giovanni Righini[*]

Macro-molecules can be designed to have an inhibitory effect on pathogens or in general to achieve maximum likelihood of interacting with some target receptors. One of the many criteria that can yield an estimate of how likely a macromolecule A is to interact with a macromolecule B is their complementarity in terms of hydropathy of their components (amino acids).

We study the problem of finding an antagonist string B with maximum hydropathic complementarity with respect to a given string A. We formally define the problem and we present some heuristics. Computational results are illustrated for the case of Semaphorin-3A.

F2C.4

OR approaches for domain prediction in proteins: methods, assessment

Piotr Łukasiak[*] *, Jacek Błażewicz, Maciej Antczak, Grzegorz Palik, Maciej Miłostan*

Current progress in computer science goes hand in hand with revolutionary advances in molecular medicine, that are generating an unparalleled increase in knowledge about human diseases.
The operations research is probably one of the most successful field of applied mathematics used in economics, physics, chemistry, almost everywhere where one has to analyze huge amounts of data. Lately, these techniques of operations research were introduced in biology, especially in the protein analysis area to support biologists.

Recent explosion of protein sequence data from all three kingdoms of life, the archea, prokarya and eukarya, gave a clear view that only because of the modular nature of proteins, and the complex ways in which their functional and structural units, protein domains, are conserved and recombined during evolution, scientist can follow such explosion and predict function of protein for which three-dimensional structure is unknown. Domains are thermodynamically stable and fold independently within the context of the whole protein. Novelty in protein function often arises as a result of the gain or loss of domains, or by re-shuffling existing domains along the linear amino acid sequence.

The prediction of protein domains is a crucial task for functional classification, homology-based structure prediction and structural genomics. Being able to solve such problem one can revolutionize biology, medicine, chemistry etc. There are many known methods designed exactly to solve the problem but none of them proved its superiority. Proposed protein domain prediction approach, is based on the application of template alignment in combination with pattern searches application. Methodology allows for rapid screening for potential domain regions and provides necessary specificity for selecting significant hits. It allows for identification domain regions without solved structural templates if sequence family definitions exist. All data needed to speed up the prediction process are stored in a specially designed database ((designed based on all available domain classification databases, e.g.: Dali Domain Dictionary, CATH, SCOP, Pfam).

Proposed method have been tested on protein chains from PDB and proved that proposed approach can be successfully applied for the considered problem.

S1A.1

Intermediate non-feasible solutions with extra capacity to solve the Double TSP with Multiple Stacks

*A. Felipe, M.T. Ortuño and G. Tirado**

The Double Traveling Salesman Problem with Multiple Stacks (DTSPMS) is a vehicle routing problem with pickup and delivery that consists on finding the minimum total length tours in two separated networks, one for pickups and one for deliveries. It is required to send one item from each location in the pickup network to the corresponding location in the delivery network. Collected items can be stored in the container in several LIFO stacks with a fixed capacity, so that every stack behaves individually as a LIFO system. There are no mutual constraints between two different stacks, but repacking is not allowed along the way.

The problem can be solved by exact methods up to 13 customers, so heuristics are needed to approach realistic sized instances. Tabu Search, Simulated Annealing and Variable Neighborhood Search have been successfully applied to the problem, being the best results obtained so far provided by the latter one. The introduction of intermediate non-feasible solutions using stacks with extra capacity diversifies the search process and improves the performance of heuristics significantly. In this work we present an algorithm to transform, if possible, a non-feasible solution using extra capacity into a feasible one, rearranging its stack assignment but keeping the routes untouched. Computational results showing that the VNS heuristic applied to the problem performs much better after the introduction of intermediate non-feasible solutions are reported.

S1A.2

Efficiently solving school bus routing problems by integrating metaheuristics and exact methods

Kenneth Sörensen[*]*, Patrick Schittekat, Marc Sevaux, Johan Springael*

The school bus routing problem discussed in this talk, is similar to the standard vehicle routing problem, but has several interesting additional features. In the standard VRP all stops to visit are given. In our school bus routing problem, we assume that a set of potential stops is given, as well as a set of students that can walk to one or more stops. The goal of this routing problem is to select a subset of stops that will actually be visited by the buses, determine which stop each student should walk to and develop a set of tours in order to minimize the total distance traveled by all buses. Evidently, this problem is NP hard.

In this talk, we discuss a metaheuristic that we developed to solve large instances of this problem. The metaheuristic is a combination of a greedy randomized adaptive search procedure (GRASP) that is used in a construction phase, and a variable neighborhood search (VNS) improvement heuristic. Moreover, the metaheuristic also integrates an exact method to solve a subproblem.

We show that the proposed metaheuristic is able to efficiently solve large-scale instances of the school bus routing problem.

S1A.3

Formulations for school bus routing problems

Emrah DEMİR, İmdat KARA*

Vehicle Routing Problem (VRP) has many applications in transportation planning, especially, in distribution and logistic. A special case of VRP is named as, School Bus Routing Problem (SBRP) where route of daily tours of buses carrying the students to the schools are determined. In SBRP, the students are picked up from gathering points and brought to school or vice versa by minimizing total transportation costs. The first published research on the SBRP was done by Newton and Thomas in 1969. Since then, large number of studies has been conducted on the subject, so then several approaches can be seen in the literature depending on the situation which is being observed.

Improvements in computing technology showed that optimum solutions of mathematical models of real life problems with moderate size can be obtained within a reasonable time. Based on this observation, we reviewed mathematical models for SBRP and proposed new formulations for symmetric and asymmetric distance and/or cost matrix, separately. For the asymmetric case, depending upon morning and afternoon problems, two node based and two flow based decision models are introduced. In order to see the computational performance of the proposed formulations, test problems taken from the literature as well as randomly generated problems are solved by using proposed models and the one recently appeared in the literature. Solutions then analyzed comparatively in terms of solution times and linear programming relaxations. It is computationally observed that, the flow based formulation outperforms all other formulations.

Keywords: Vehicle Routing Problem, School Bus Routing Problem, Integer Programming Model

S1A.4

A 1-ShifT Local search Heuristic with low complexity for probabilistic traveling salesman problem

*Alejandro Lamas, Pablo A. Miranda**

Probabilistic Traveling Salesman Problem (PTSP) is a generalization of the well studied Traveling Salesman Problem (TSP). PTSP considers that the nodes have a fixed, known probability to require a visit. Consequently, not all nodes must be visited in the same tour or period. The objective of PTSP is to determine a visit sequence to all nodes, minimizing expected costs and considering all problem realizations. A realization of the PTSP is a subset of nodes that requires a visit in the same period. PTSP is a NP-Complete problem that implies a high resource consumption in order to solve the problem.

Given the high complexity in the resolution of PTSP, this problem has been addressed by using heuristic approaches. Generally, these heuristics are adaptations of TSP local search algorithms. Examples of these heuristics are 2-opt and 1-shift heuristics. In contrast to the TSP, there exists a high complexity in the associated saving computation of movements within the neighborhood defined by the heuristics. This high complexity is due to the large number of realizations that must be considered in this computation.

The focus of this work is to present a 1-shift heuristic to PTSP with homogeneous visit probability, where the exact saving movement is not considered as a goodness criterion, but only realizations with high occurrence probability. The set of all considered realizations in the goodness criterion does not depend on the instance size (number of nodes). This criterion only depends on instance probabilities. Then, the proposed heuristic has a complexity lower than the standard 1-shift heuristic applied to PTSP.

Heuristic design requires to define the number of realizations that must be considered in computation saving, by the application and demonstration of two concepts: (1) the asymptotic behavior presented by the probability associated to a subset of realizations, as the number of nodes increases and

(2) this probability does not affect strongly the quality of the obtained solutions, compared to those obtained by traditional 1-shift heuristic applied to PTSP.

S1B.1

A modelling of animal selection scheme optimized by the genetic algorithm

A.D. Costard[*]*, Z.G. Vitezica, C. R. Moreno, J.-M. Elsen*

Farm animal diseases might represent a risk to human health. Currently, different strategies of prevention, or treatments are used to fight against these diseases. But when a major gene controlling the susceptibility or the resistance to the disease is known, the genetic selection can be a very efficient additional strategy. The aim of this selection is to decrease the numbers of affected animals until the disappearance of the disease. Because of existing selection schemes on traditional production traits for most of the farm animals, the genetic strategy have to take into account them. So this strategy becomes the simultaneous selection on two traits, a quantitative trait underlying a polygenic variation (the production trait) and a monogenic trait (i.e. resistance to the disease). A mathematic approach was developed to model and optimize this dual purpose selection. A traditional selection scheme taking into account overlapping generations, different selection strategies for males and females, and assortative mating was modeled. We propose a deterministic model which allows to perform a global optimization of the selection scheme in order to maximize the frequency of the desired genotype for the monogenic trait while minimizing the loss of genetic progress on the polygenic trait. Genetic Algorithm was used to solve this optimization problem. This method can be used to develop and evaluate different optimal strategies in order to obtain applicable strategies in farms. This model was applied to the example of selection for the scrapie resistance gene: *Prnp* gene. This gene was introduced as an additional selection criterion in an already existing dairy sheep selection scheme.

S1B.2

Genetic algorithms with local search for the unrelated parallel machine scheduling problem with setup times

Eva Vallada[*]*, Rubén Ruiz, Dario Diotallevi*

In the unrelated parallel machine scheduling problem there is a set $N = 1,.., n$ of n jobs that have to be processed on exactly one machine out of a set $M = 1..m$ of m machines. Machines are considered unrelated when the processing times of the jobs depend on the machine to which they are assigned to. This is the most realistic case which is also a generalization of the uniform and identical machines cases. The most studied optimization criterion is the minimization of the maximum completion time of the schedule, a criterion that is known as makespan or *Cmax*. Parallel machine scheduling problems have been widely studied in the past decades. However, unrelated parallel machines have been much less studied. Additionally, the consideration of sequence dependent setup times between jobs has not been considered until recently. In the literature, we can find several heuristic and metaheuristic algorithms for the parallel machine scheduling problem considering setup times. In this paper, we deal with the unrelated parallel machine scheduling problem in which machine and job sequence dependent setup times are considered. We review, evaluate and compare the methods available in the literature. We also propose a new genetic algorithm that shows excellent performance for a large benchmark of instances.

In our proposed algorithm, the initial population is randomly obtained except one individual which is given by the multiple insertion heuristic (MI). The most innovative features of the algorithm are the crossover operator and the local search. The first one includes a small local search procedure to obtain good offspring. After the crossover, mutation is applied according to a probability and a very fast local search procedure is applied to the offspring. The local search is based on the insertion of all the jobs in every position of all the machines. When a job is inserted in some machine it is not necessary to evaluate again the full sequence to obtain the new makespan value, so the local search is really fast. After that, the offspring are inserted into the population only if they are better than the worst individual of the population and if at the same time there are no other identical individuals already in the population. We carried out a complete calibration experiment of the different parameters and operators of the proposed algorithm by means of a design of experiments. We obtained two genetic algorithms denoted as GA1 and GA2, respectively. We compared the results with the following methods available in the literature: the seventh heuristic (the best performing one) proposed by Weng et al. (2001), denoted as HWeng and the Multiple Insertion heuristic according to Kurz and Askin (2001), denoted as MI. Regarding the metaheuristics, we have re-implemented the tabu search algorithms presented by Franca et al. (1996) and by Logendran et al. (2007) (denoted by TS and TS_L, respectively). Moreover, a genetic algorithm, proposed by Kurz and Askin (2001) (denoted as GA), a simulated annealing proposed by Low (2005) (denoted as SA), a heuristic proposed by Rabadi et al. (2006) (denoted as Meta) and a GRASP algorithm presented by Armentano and Felizardo (2007) (denoted as GMC) have been also implemented. To test all the methods we propose a new benchmark of instances which is formed by 640 problems ranging from 6 to 12 jobs and from 2 to 5 machines (small instances) and 1000 problems ranging from 50 to 250 jobs and from 10 to 30 machines (large instances). The stopping criterion for the metaheuristic methods is set to a maximum elapsed CPU time according to the number of jobs/machines. The table shows the results by means of the average relative percentage deviation (RPD) over the best known solution for the different algorithms (small and large instances).

	HWeng	MI	GA	Meta	SA	TS_L	TS	GMC	**GA1**	**GA2**
RPD (small)	40.63	19.43	8.36	6.27	1.83	5.22	23.46	7.84	2.71	**1.38**
RPD (large)	49.39	36.9	432.2	48.78	36.18	29.19	65.49	64.28	8.87	**5.75**

As we can see, the proposed algorithms (GA1 and GA2) produce excellent results, across all instances, especially the second one (GA2). If we focus on the large instances, we can see that the proposed algorithms outperform, by a huge margin, the remaining methods.

In this paper we have proposed a genetic algorithm for the unrelated parallel machine problem with setup times. The proposed method results in a much superior performance when compared to existing alternatives, especially for large instances.

S1B.3

Maximizing Revenue on Parallel Machines by Simulated Annealing Method

Malgorzata Sterna*, Jacek Juraszek, Erwin Pesch

The work concerns the problem of maximizing revenue from executing a set of jobs on a set of parallel identical machines. Each job is described by the release time at which it becomes available for processing and the execution time for which it has to be performed without preemption. Moreover, the revenue is given for each job representing income from its non-delayed execution. Finishing an activity before its due date results in the full revenue. If a job is executed after its due date, the revenue is decreased by the penalty factor equal to the weighted tardiness. In a feasible solution, the completion time for a particular job cannot exceed its deadline. The goal is to select a subset of jobs and to schedule them in the feasible time intervals between release times and deadlines on the set of identical parallel machines in order to maximize the total revenue.

The presented problem models many practical situation, when a decision-maker has to decide whether to accept or to reject customer orders taking into account possible income for a company and the capacity of its resources.

Since the scheduling problem on a single machine with the weighted tardiness criterion is already unary NP-hard, the case under consideration is also intractable.

We propose the simulated annealing method (SA) for solving the problem under consideration. The solution is represented as the assignment of jobs to machines. The sequence of a subset of jobs on a particular machine is determined by the list scheduling approach. The rejected jobs are allocated on an additional virtual machine, for which the execution order is arbitrary. The simulated annealing algorithm starts with a heuristic schedule and interchanges or shifts jobs between machines depending on the neighborhood definition. SA is additionally equipped with the intensification mechanism – a local search algorithm for improving a current solution, and the diversification mechanism randomly fluctuating it in the case of obtaining a local optimum.

Computational experiments were performed for a set of randomly generated input instances of various characteristics. They differed – in order to vary their difficulty – in the release times and due dates of jobs as well as the number of jobs per machine factor. We tested the exact method, the list scheduling heuristic and the simulated annealing metaheuristic, comparing their time efficiency and the quality of generated solutions.

S1B.4

What's the best way to implement Simulated Annealing?

Mike Wright

Simulated Annealing (SA) is a metaheuristic neighbourhood search technique for solving combinatorial optimisation problems. Its use has been widespread for many years in a very large number of fields of study for solving all sorts of problems, both theoretical and practical. Google Scholar (as of 4th March 2008) detects 7,550 articles with the words "simulated annealing" in the title, and 167,000 which contain the phrase somewhere in the article. Given its popularity and success, it is therefore surprising that there appears to have been no systematic approach to determining the "best" way to use SA.

In order to run SA, a cooling scheme and certain parameters must be prespecified – a starting temperature, a mechanism for changing this temperature and a stopping criterion, which may be related to temperature, time taken, number of iterations, success measures, etc. Some authors in their published work describe experimental fine-tuning of these parameters before SA is applied; others proclaim some principle behind their choice (e.g. "the initial temperature must be such that *x*% of worsening perturbations are accepted", though the value of *x* chosen tends to vary between authors); other authors are vague on the subject or fail completely to justify, or even in some cases to mention, their choices of cooling scheme and parameters.

What is clear is that there is no consensus as to what works best. This paper therefore attempts to tackle this question.

Of course, in common with all neighbourhood search methods, better results can always be produced by simply letting the technique run for longer times. In the case of SA this can be easily achieved by reducing the speed of cooling. Therefore, in order for meaningful comparisons to be made, it must be assumed that there is a fixed amount of time, or of computational effort, available.

This paper reports on experiments undertaken to compare the effects in terms of solution quality of different sets of parameters for a randomly-generated example of the Traveling Salesman problem (TSP) with 100 cities. This is of course a simple problem in structure, but *np*-hard and large enough to ensure that it is by no means easy to reach a solution which is optimal or very close to optimal. This makes it a suitable vehicle for experimentation.

The cooling scheme used for these experiments is a simple non-dynamic geometric method, where at every iteration the temperature is multiplied by a fixed constant, with no reheating or restarting allowed. The initial temperature and final temperature is prespecified, as is the number of iterations (a suitable proxy for time taken). The cooling constant can be easily calculated from these.

This is not to suggest that it may not be possible for better results to be achieved using more sophisticated approaches, for example with a dynamic temperature control mechanism, but it seems sensible to start our experimental work with a simple form of SA.

The experiments will determine the best combination of starting and ending temperatures to use for this particular problem, depending on the number of iterations available (a good proxy for time available) and whether the starting solution is generated randomly or by means of a nearest neighbour method. These temperatures then need to be normalised as they are not dimension-free; this is done by translating them into acceptance percentages at the start and end of the process, to enable the results to be properly interpreted in the context of different problems.

Since SA is a stochastic technique, a very large number of experimental runs are required to ensure that the outcomes are statistically significant at levels of confidence close to unity. The experiments therefore will have been run over a period of several months.

The results of these experiments will be presented at the conference and their implications will be discussed. Ideas for future research will also be put forward.

S1C.1

The inverse Fermat-Weber problem

Mohammadreza Galavii[*], Rainer E. Burkard, Elisabeth Gassner

Given n points in the plane with nonnegative weights, the inverse Fermat-Weber problem consists in changing the weights at minimum cost such that a pre-specified point in the plane becomes the Euclidean 1-median. The cost is proportional to the increase or decrease of the corresponding weight. Formulated as linear program the inverse Fermat-Weber problem can be solved in linear time. We derive a purely combinatorial algorithm which solves the inverse 1-median problem in a plane under Euclidean or squared Euclidean metric with unit cost by a greedy-like algorithm.

S1C.2

Inverse Network Center Location Problems with Edge Length Modifications

Behrooz Alizadeh[*], Rainer E. Burkard

Network center location problems are basic models in operations research and play an important role in practice and theory. While in classical network center location problem the goal is to determine the optimal location of a facility such that maximum of the (weighted) shortest distance to all customers is minimized, the inverse network center location problem is concerned with modifying the parameters of a given network, like edge lengths or vertex weights, at minimum total cost such that a pre-specified point becomes a center of the network. In this talk we consider the inverse center location problem on a tree network in which we are allowed to increase or decrease the edge lengths within certain bounds. If all costs are equal, then a quadratic time algorithm for the problem under investigation is developed. For arbitrary costs, a cubic time algorithm is designed and analyzed. Finally, it is shown that the problem is solved in linear time, when the network is a path or star graph.

Keywords: center location, inverse optimization, combinatorial algorithms

S1C.3

Inverse Median Location Problems with Point Coordinates Modification on a d-Dimensional Real Space

Fahimeh Baroughi, Rainer E. Burkard , Behrooz Alizadeh*

In this talk we investigate the inverse p-facility median problem on a d-dimensional real space in which the task is to modify the coordinates (or weights) of n existing points at minimum cost within specific bounds so that the set of p prespecified points becomes a p-facility median. First we consider the inverse 1-median location problem and develop O(dn)-time solution methods for this problem, provided that the distances are measured in rectilinear norm or squared Euclidean norm. Moreover, it is shown that a special inverse p-facility median problem under squared Euclidean norm can be transformed to a linear optimization problem which is solvable in O(dn)-time. If instead of the point coordinates variations we consider the modification of point weights, then the inverse p-facility median problem can be solved in O(pn)-time.
Keywords: location problems, 1-median, p-median, inverse optimization

S1C.4

Upper Bounds for Discrete Competitive Facility Location Problems

V. Beresnev

We consider the facility location problem in the presence of competition, when two competitive firms open facilities sequentially to maximize own profits. Each client selects one of the open facilities according to his preferences and gives a corresponding profit to the first firm (Leader) or the second (Follower). The problem is to find a facility location for the Leader which maximizes its profit taking into account the optimal reaction of the Follower. We consider two problem formulations which differ in the Follower's objective function. In the first problem Follower tries to maximize own profit and in the second one the Follower wishes maximize the total number of the clients captured. We use fixed costs of the opening facilities and formulate models as a bilevel linear integer programming problems. We present equivalent formulations of these problems in the form of the bilevel pseudo-Boolean programming. Using these representations and some properties of the optimal solutions for the Follower problem, we construct auxiliary pseudo-Boolean function. The minimum of this function leads us to an upper bound for optimal values of the Leader's profits. Computational results illustrate the good performance of the upper bound for test instances which optimal solutions are known.

S2A.1

Scheduling reentrant jobs on parallel machines with a remote server

K. Chakhlevitch, C.A. Glass*

We explore a specific combinatorial problem relating to re-entrant jobs on parallel primary machines, with a remote server machine. A middle operation is required by each job on the server before it returns to its primary processing machine. The objective is to schedule operations of the jobs on the primary machines and on the server so as to minimize a makespan, i.e. the latest completion time of any job. The problem is inspired by the logistics of a semi-automated micro-biology laboratory. The testing programme in the laboratory corresponds roughly to a hybrid flowshop, whose bottleneck stage is the subject of study. We demonstrate the NP-hard nature of the problem, and provide various structural features. A heuristic is developed and tested on randomly generated benchmark data. Results indicate solutions reliably within 1.5% of optimum. We also provide a greedy 2-approximation algorithm. Test on real-life data from the microbiology laboratory indicate a 20% saving relative to current practice, which is more than can be achieved currently with 3 instead of 2 people staffing the primary machines.

S2A.2

Iterated Greedy Algorithms for the Unrelated Parallel Machine Problem

Luís Fanjul, Rubén Ruiz, Michele Ciavotta*

This work deals with the assignment and sequencing of a set of *n* jobs that have to be processed on exactly one out of *m* parallel machines. No machine can process more than one job at the same time. The most general case is when the processing time depends on the machine to which the job is assigned to. This case is known as the unrelated parallel machine problem. Therefore, we denote by p_{ij} the non-negative, non pre-emptive, processing time of job *j* at machine *i*. In general, there are m^n possible solutions to this problem which has been shown to be *NP*-Hard in the strong sense. In the scheduling area, this problem is denoted by $R//C_{max}$. Many algorithms and methods have been proposed for this hard combinatorial problem, including many highly sophisticated methods. By contrast, in this paper, we propose a simplified metaheuristic that produces solutions of extreme quality that are comparable, or even better, than the current state-of-the-art.

The initial solution is obtained constructively by inserting each job, in order, into the machine that increases the completion time the least. Iterated Greedy (IG) is a very simple method that, starting from a complete solution, iterates over two operators: destruction and reconstruction. With destruction, some elements of the initial complete solution are removed which are later greedily re-inserted into the solution in the reconstruction operator. IG was initially proposed for the flowshop scheduling problem by Ruiz and Stützle (2007) with great success. For the unrelated parallel machine problem some assigned jobs are randomly removed in the destruction operator. In the reconstruction, the removed jobs are inserted in the machine in which they increase the completion time the least. Additionally, we apply, after reconstruction, a VNS local search phase using the insertion and exchange neighbourhoods. The local optimum is accepted as the new solution only if it improves the current best. Initial experiments indicated that destroying, at each iteration, 2% of the jobs, resulted in the best solutions.

The proposed IG method is compared against the following algorithms: MIP model solved with the last version of CPLEX 11.0. RBS method by Ghirardi and Potts (2005) and PARTIAL by Mokotoff and Jimeno (2002). The last two methods are currently considered state-of-the-art with the best solutions known in the literature for this problem. The benchmark used for the computational evaluation is composed by 200 instances ranging in size from 100 jobs and 10 machines to 1000 jobs and 50 machines. The processing times are uniformly distributed between 100 and 200. All algorithms are tested in a PC/AT computer with a 3GHz Pentium IV processor and 1 GB or RAM memory. CPLEX and IG algorithms are stopped after 15 seconds of CPU Time. For each algorithm, we measure the relative percentage deviation from the best solution known and the CPU time used is given between parentheses. The following table shows the obtained results:

IG	CPLEX	RBS	PARTIAL
0.27 (15)	0.51 (15)	0.56 (102.20)	0.78 (22.49)

Although not shown due to reasons of space, an ANOVA analysis is carried out on all results, showing statistically significant differences between all averages except CPLEX and RBS. As we can see, the IG method, which is remarkably simple, obtains results that are, on average, 46% better than its close competitor, CPLEX 11.0. An additional interesting result is that a regular commercial solver like CPLEX, shows better results than ad-hoc specific algorithms like RBS or PARTIAL. Furthermore, this latter algorithm uses CPLEX as a subrogate method. Lastly, the CPU times for the proposed IG method are forced to be less or equal than those of its competitors.

In this paper we have proposed an effective, yet simple Iterated Greedy algorithm for a classical combinatorial problem in the scheduling area: the unrelated parallel machine problem. The main contribution of this paper is in the form of a new algorithm that is much easier to code than all competitors but at the same time produces much better results in less CPU time. Additionally, we have shown that regular MIP models solved with CPLEX produce better results than specific algorithms. This is especially true for the later versions of CPLEX (11.0). Future research includes further developments on the local search phase as well as applications to other parallel machine problems.

S2A.3

Due date assignment and just-in-time scheduling on uniform parallel machines

Nguyen HUYNH TUONG and Ameur SOUKHAL[*]

Known as Just-In-Time scheduling problems, they are related to real industrial applications. This study deals with minimizing the total weighted earliness-tardiness and common due date assignment on *m* uniform parallel machines. *n* independent jobs should be completed without preemption as close as possible to common due date. All jobs are ready at time zero. This problem is known to be NP-hard even in the case of single machine and a given due date. However, well solvable cases are identified. Particularly, the two uniform parallel machines with identical jobs (equal job earliness/tardiness penalties and identical processing times) studied, recently, by Mosheiov and Sarig (2008). They show that the $Q2|p_i=p,d_i=d|(\sum aE_i+bT_i+cd)$ is polynomially solvable in a constant time. However, in the case of *m* machines, three following conditions should be analysed on each machine:

(i). there is no idle time at the beginning and there is a "just-in-time" job (i.e. the job does not have any earliness-tardiness penalty);

(ii). there is no idle time at the beginning and no "just-in-time" job;

(iii). there are an idle time at the beginning and a "just-in-time" job.

Hence the number of cases to be checked is bounded by 3^m. For each of these cases, the optimal schedule and due date are found in a similar procedure defined in the case of two uniform machines. Note that this effort increases very fast with the number of machines.

In our study, we propose another procedure to solve efficiently the general scheduling problem $Q|p_i=p,\ d_i=d|(\sum aE_i+bT_i+cd)$ (i.e. *m* is not fixed). We show that an optimal solution can be found in $O(m^3)$ time which is better than $O(3^m)$ proposed by Mosheiov and Sarig.

Keywords: Just-In-Time Scheduling; Uniform Parallel Machines; Identical Processing Times; Polynomial Cases.

S2A.4

Solving a special case of the discrete-continuous scheduling problem by metaheuristics and linear programming approach

Rafał Różycki

The problem of scheduling non-preemptable jobs on parallel identical machines with an additional renewable and continuously divisible (continuous) resource is considered. The problem belongs to the class of discrete-continuous scheduling problems. In the considered problem, processing rate of a job depends on the amount of the continuous resource allotted to this job at a moment. This relation is described by the known non-decreasing convex function. Each job requires for its processing a machine and an amount of continuous resource. A job is performed by at most one machine at a moment. Moreover, each job is characterized by the following parameters:

- processing demand (related to the size of a job),
- required minimal amount of the continuous resource,
- acceptable maximal amount of the continuous resource.

The processing time of a job is unknown in advance and it depends on the allocation of the resources. As a scheduling criterion we assume the total length of a schedule– Cmax.
The proposed special case of the discrete-continuous scheduling problems may be useful for discovering an optimal fire–fighting strategy. The fire of building can be modeled as a set of jobs of given initial sizes, which compete for scarce renewable resources: water (continuous resource) and set of fire-hoses (parallel identical machines).
The general methodology for solving the problem is proposed. The methodology bases on the decomposition of the problem into two interrelated subproblems: the problem of assigning jobs to machines and the problem of allocating of the continuous resource to jobs already allocated on machines. A formulation of linear programming problem is proposed for finding an optimal continuous resource allocation to jobs already sequenced on machines. Unfortunately, the number of feasible sequences of jobs on machines grows exponentially with the number of jobs. Thus, it is justified to utilize a metaheuristic approach to find an optimal solution of the problem.

S2B.1

Taming Wind Energy with Battery Storage

Andreas Ernst and Gaurav Singh[*]

The use of wind to generate electrical energy is becoming more popular around the world as global efforts are made to deal with green house gas emissions from more traditional sources of energy. In Australia, wind energy is one of the technologies being promoted by mandatory renewable energy targets set by the government. In comparison to other renewable energy sources, wind energy is more economical and eco-friendly, but it has one significant problem. Its electricity production is inherently highly variable and difficult to predict. This variability causes problems at various time scales. Over longer time scales it is difficult to match electricity generation to daily and seasonal patterns of demand. On shorter time scales high frequency "noise" in electricity output causes problems for network stability and managing the short term dispatch of generators to meet demand.

Most of the previous work in this area concerned battery-management. Here we focus on smoothing the output over short time periods in order to improve network stability. The CSIRO is developing a prototype system in Australia that will use specially developed batteries to store some of the wind energy. The aim is to use a relatively small amount of storage in order to smooth the wind energy output, making it more predictable and reliable in the short term. As wind farms are becoming larger and satisfy a more significant proportion of overall Australian electricity demand this is becoming a more significant issue.

A battery can be located at a wind farm. This is often appropriate when it is located in a remote area and all of the variability has to be handled by a single transmission link. Alternatively, a battery can be located at a critical part of the network such as the end of a transmission link which connects multiple wind farms. The models and algorithms presented in this paper deal with both cases.

We consider the following two variants of the problem: (1) how much battery capacity is required to provide a certain level of reliability? and (2) for a given battery size what are "good" control algorithms for the battery?

We present linear programming formulations for each of these problem variants and compare their performance against online algorithms and descent methods. We also present estimates of the minimum battery capacity required depending on the ramp rate limit using these methods.

Keywords: wind farming, battery storage, online algorithms, control methods, linear programming

S2B.2

Reoptimization techniques in a column generation scheme

Nora Touati,Lucas Létocart, Anass Nagih*

Reoptimization approaches are used successfully to improve iterative algorithms, like subgradient methods. We are interested in their performances in a column generation scheme. We propose in this work a reoptimization technique which permits to solve efficiently the subproblem at an iteration k of column generation, by using some information about the resolution of the subproblem at iteration k-1.

Principle of reoptimization:

This method is used to solve efficiently a sequence of different instances of a problem. We denoted by P1 and P2 two different instances of a problem, i.e P1 and P2 have the same structure, but different data. Reoptimization consists of studying possibilities of resolution's acceleration of P2, using some information about the resolution of P1.

Reoptimization technique in a column generation scheme:

At each iteration of column generation, a pricing problem is solved. Pricing problems differ only by their objective coefficients, so, we can exploit the resolution of a pricing problem at an iteration to solve efficiently the pricing problem at the next iteration.

At the last iterations of column generation, master problem's values are close. Pricing problem's solutions can then be neighbors, in this case reoptimization can reduce significantly the time of pricing problem's resolution.

The effectiveness of this approach is demonstrated by computational experiments on two kinds of instances, a number of Solomon's test problems and some generated instances of the vehicle routing problem with time windows.

S2B.3

Using "hyper-tubes" for maintaining diversity in Multi-Objective Particle Swarm Optimizer

Mario Alberto Villalobos-Arias

In the last few years, several multi-objective particle swarm optimizers (MOPSOs) have been proposed in the specialized literature (see for example [1, 6, 2]). However, one of the key issues to successfully extend particle swarm optimizers to handle multiple objectives is to use an effective mechanism to maintain diversity. In some recent work, a few authors have proposed or adopted novel mechanisms to maintain diversity in their MOPSOs (e.g., [5, 4]). In this presentation, we propose a new mechanism to maintain diversity, which we show to overcome the main drawbacks of other popular mechanisms such as ε-dominance [3] and the sigma method proposed by Mostaghim [5].

Keywords: multi-objective, optimization problem, particle swarm, heuristic.

References

[1] Carlos A. Coello Coello, Gregorio Toscano Pulido, and Maximino Salazar Lechuga. Handling Multiple Objectives With Particle Swarm Optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):256–279, June 2004.

[2] Xiaohui Hui, Russell C. Eberhart, and Yuhui Shi. Particle Swarm with Extended Memory for Multiobjective Optimization. In 2003 *IEEE Swarm Intelligence Symposium Proceedings,* pages 193–197, Indianapolis, Indiana, USA, April 2003. IEEE Service Center.

[3] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. *Evolutionary Computation,* 10(3):263–282, Fall 2002.

[4] Sanaz Mostaghim and Jürgen Teich. The role of "ε-dominance in multi objective particle swarm optimization methods. In *CEC'2003,* volume 3, pages 1764–1771. IEEE Press, December 2003.

[5] Sanaz Mostaghim and Jürgen Teich. Strategies for Finding Good Local Guides in Multiobjective Particle Swarm Optimization (MOPSO). In *2003 IEEE Swarm Intelligence Symposium Proceedings*, pages 26–33, Indianapolis, Indiana, USA, April 2003. IEEE Service Center.

[6] Sanaz Mostaghim and Jürgen Teich. Covering Pareto-optimal Fronts by Subswarms in Multi-objective Particle Swarm Optimization. In *CEC'2004,* volume 2, pages 1404–1411. IEEE Service Center, June 2004.

S2B.4

Circular chains of Chinese dice

Eduardo Piza[*], *Leo Schubert*

In this paper we study the *Chinese dice*, mathematical objects similar to the ordinary dice but allowing repetition in its face values. We say that a die *A is preferred over* a die *B* (written $A < B$) if *A* wins more frequently than *B* do.

We study first the existence of a circular chain of three dice *A*, *B*, *C* such that $A < B < C < A$, using a mixed integer programming algorithm. Then we generalize the problem to *n*-dimensional dice—that is, dice of *n* faces (with $n \geq 4$)—and we search circular chains of length *m* (with $m \geq 3$) by using a simulated annealing algorithm.

We compare some different objective functions and also obtain good solutions to the problem with very efficient algorithms. Finally we obtain a theoretical result concerning the existence of general circular chains.

Keywords: Chinese dice, mixed integer programming, simulated annealing, combinatorial optimization.

S2C.1

Lower bounds and flat graphs of precedence relations for the resource-constrained project scheduling problem

Evgeny R. Gafarov, Alexander A. Lazarev[*]

We consider some special cases of the *NP*-hard resource-constrained project scheduling problem *RCPSP* to minimize the makespan. We propose a solution algorithm based on the *Ant Colony Optimization* meta-heuristic, which was incorporated in commercial software program *1C* to handle practical project scheduling problems with resource constraints.

We show that well-known lower bounds [Brucker P., Knust S.; Complex Scheduling. Springer:Berlin; 2006] for the problem may yield bad approximation ratios or their calculation is an *NP*-hard problem. We prove that the calculation of the lower bound of Mingozzi et al., LB_M, is an *NP*-hard problem and there exists an instance of *RCPSP* for which $C^*_{max}/ LB_M \approx 2$.

We conjecture that the ratio of the optimal makespan of *RCPSP* to that of the preemptive version of the problem is less than 2.

For the case A (with a single resource (Q_1 units) and an empty graph *G*, i.e., $e = 0$) we have the following relation: $C^*_{max} - p_{max} < 2\ C^*_{max}\ (pmtn.)$.

For several resource-constrained project scheduling problems with precedence relations, we have the following theorem.

Theorem. For any instance of *RCPSP* with *n* jobs and *e* relations, there exists an analogous instance with a flat graph *G'* with *n'* jobs and *e'* relations, where $n + e \leq n' + e'$.

We obtain an analogous instance from the original one by adding "dummy" jobs and deleting all the unnecessary relations.

S2C.2

Algorithms for the Special Case of the Total Tardiness Problem on a Single Machine

Alexander A. Lazarev

In the paper, we study NP-hard in the ordinary sense scheduling problem of minimizing total tardiness on a single machine $1|\ |\sum T_j$. We introduce four algorithms both polynomial and pseudo-polynomial for finding an optimal schedule for all sub-cases of the case when $p_1\geq p_2\geq\ldots\geq p_n$ and $d_1\leq d_2\leq \ldots\leq d_n$. The processing time p_j and due date d_j are given for all jobs from N. The machine processes only one job at a time without preemptions.

A schedule π is defined as a permutation of elements of the set N. The problem minimizing total tardiness calls for a construction of a schedule π * that minimizes the total tardiness function

$F(\pi)=\sum \max\{0, C_j(\pi)- d_j \}$. The value $C_j(\pi)$ is called the completion time of the job j in a schedule π.

To classify and solve the sub-cases of this case, we introduce the approach when the set N of jobs is partitioned into $\mathbf{k}\leq n$ subsets and then an optimal schedule is constructed based on this partition. The procedure runs in O(n) time.

```
Partitioning Procedure
$\mathbf{k}:=1$, $\alpha_1:=1$;
FOR $j=2,3,\ldots,n$
 IF $d_j-d_{\alpha_{\mathbf{k}}}>p_j$
   THEN $\beta_{\mathbf{k}}:=j-1$; $\mathbf{k}:=\mathbf{k}+1$; $\alpha_{\mathbf{k}}:=j$;
 ENDIF
ENDFOR
$\beta_{\mathbf{k}}:=n$; $M_\nu=\{\alpha_\nu,\alpha_{\nu+1}\},\ldots,\beta_\nu\}$, $\nu=1,2,\ldots,\mathbf{k}$.
$N=M_1U\ldots U M_{\mathbf{k}}$.
```

Algorithm B-**k** constructs an optimal schedule for an instance with an arbitrary **k** in $O(\mathbf{k}n\sum p_j)$ time. Algorithm B-1 is used to solve the sub-case when **k**=1 in $O(n \sum p_j)$ time. If **k**=1 and $d_n-d_1\leq 1$, then Algorithm C-1 finds an optimal schedule in $O(n^2)$. And finally, if **k**=n, then we can use Algorithm B-n to construct an optimal schedule in $O(n^2)$ time.

We also study Algorithm B-1 with purpose to find cases for which the algorithm constructs an optimal schedule. As the result, we prove that Algorithm B-1 solves the instances of the NP-complete even-odd partition problem too.

S2C.3

Some graph issues for a sequenced traffic flow

D. Prot, C. Rapine, S. Constans, R. Fondacci*

The present air traffic management (ATM) system used in Europe is based on a route network, which means that aircraft do not fly on their orthodromic route, that is to say directly from their origin to their destination. It implies a 5.9% route lengthening compared to the use of orthodromic routes. Estimated extra-cost to airspace users is 2 230 millions of euros, and 4.7 millions of tons of additional CO_2 are emitted. Thus it is a necessity to look at an ATM system based on orthodromic routes. The use of such routes however generates a huge number of intersections that air traffic controllers have to monitor, increasing their workload. For this reason, in this communication, we consider an alternative and very organised air traffic framework, very different from what is operated today, at the opposite of the free flight concept. On each orthodromic route defined by an origin, a destination and a given flight level, immaterial periodic moving points are periodically generated. Two types of moving points exist, the allowed ones and the forbidden ones. Aircraft are compelled to follow allowed moving points. The problem is then to determine which Origin/Destinations (ODs) can be put on the same flight level without generating a conflict (loss of separation norms) between two aircraft, or, at most, very few. Modelling this problem with convenient graphs highlights a link to that of graph colouring. This provides theoretical bases to address some properties of our problem. More precisely, we show that it has a higher complexity than the colouring problem on specific classes of graphs. This is a critical issue since we have to deal with huge graphs having 30 000 ODs and about 2 millions of intersections over Europe.

S2C.4

Production Planning in the Interior of the Automotive Industry's Production Chain

Claas Hemig

In the last two decades automotive OEMs were forced to react on changing products' demands and installed flexible manufacturing systems to be able to adjust production time and speed, to use flexible working hours, hirings and dismissals as well as to shift production workload and workers between production facilities, typically production lines inside a plant. This flexibility is limited to technical restrictions, labor legislation and in-plant agreements, e.g. with the staff association or labor union.

An automotive plant typically consists of the body and paint shop welding and painting the car bodies, and the final assembly installing e.g. the engine, gearbox, and interior decoration. Product specific buffers of limited capacities are located between subsequent shops to decouple the corresponding production processes.

The planning horizon of four to six years is separated into T periods, typically weeks or months. Associating the periods with the stages we use a Dynamic Programming approach to find a cost-optimal solution for the outlined planning problem with L production lines and P different products. To reduce the number of decisions and states we merged similar states into one and ceased to enumerate all meaningful decisions. For this, we now focus on the subproblem of distributing the workload between the lines in the paint shop in a certain period t assuming the products' demand as given from the subsequent shop and the decisions concerning the capacities of the lines as fixed. The problem for the body shop is an analogous one.

The workload in the paint shop is determined by the production program in the final assembly and the buffer capacity lying between them. To provide a buffer-feasible production plan we have to ensure feasibility for every single shift of the considered period because typically the buffer has a capacity of about one shift.

Therefore we model the problem as a classical transportation problem where lines supply and products ask for production capacity. Buffers have to be introduced as suppliers as well as demanders. In particular, we introduce a demander L_{ls} for each line l in each shift s with a demand equal to the corresponding production capacity. In analogy we introduce a supplier D_{ps} for each product p in each shift s with a supply equal to the corresponding product's demand. To model the product specific buffers we introduce a supplier B_{ps}^{S} as well as a demander B_{ps}^{D} for each product p in each shift s. The buffer capacities—reduced by the minimum safety stock—act as supply as well as demand, while the demand for the first shift is reduced by the opening stock. To balance the transportation problem we introduce a dummy demander D.

The resulting transportation graph is rather sparse because transportation is restricted to only a few arcs as follows. In particular, each demander L_{ls} is connected to D_{ps} and B_{ps}^{D} if and only if p can be produced on l which models the production for fulfilling demand and increasing the buffer stock, respectively. Additionally, L_{ls} is connected to D for all l and s to model unused capacity. Each supplier B_{ps}^{S} is connected to B_{ps}^{D}, $D_{p,s+1}$, and $B_{p,s+1}^{D}$ to model unused buffer capacity, fulfilling next shift's demand from buffer, and storing inventory for future shifts, respectively. For the last shift the latter two arcs are replaced by an arc to D modeling the closing stock. The arcs are valued with appropriate production and storage costs.

Solving the transportation problem by the network simplex algorithm provides a production plan that can be used as a part of a decision inside our Dynamic Programming approach. This approach enables us to solve real-world instances of our industrial partner in acceptable time and quality.